BACHELOR

Analysis of the expected time to ruin in several insurance models

van der Zande, Jerom

Award date:
2021

Department of Mathematics
Stochastic Operations Research
Postbus 513, 5600 MB Eindhoven
The Netherlands
www.tue.nl

Author
J.F. van der Zande (1248960)

Supervisor
J.A.C. Resing

Date
January 29, 2021

Analysis of the Expected Time to Ruin in Several Insurance Models

Bachelor Final Project

Abstract

The main goal of this study is to analyze the expected time to ruin, both analytically and numerically, in several insurance models.
There are four main chapters in this report, each with its own unique insurance model, all consisting of a part with exact computations and a part with a stochastic simulation. The models that will be discussed are the Cramér-Lundberg model, also known as the classical Poisson risk model, a model with stochastic premium, a model with both linear and stochastic premium and in the last chapter we will look at a model with stochastic premium and an underlying Markov process with two states affecting the claim arrival rate. For the first three models, an explicit expression for the expected time to ruin has been found, using different techniques in each chapter, validated by the stochastic simulation. For the last model, no expression for the expected time to ruin has been found analytically, but by using a simulation we are still able to observe how this model behaves for different sets of parameters.

Technische Universiteit
Eindhoven
University of Technology

Table of contents

Title
Analysis of the Expected Time to Ruin in Several Insurance Models

1 Introduction 1

1.1 Outline of the report . . . 1

2 Cramér-Lundberg Model 2

2.1 The Classical Risk Process . . . 2

2.2 Time to Ruin . . . 2

2.3 Exponential Claims . . . 6

2.4 Simulation . . . 7

2.4.1 Code . . . 7

2.4.2 Results . . . 8

3 Model with stochastic premium 12

3.1 The Mathematical Model . . . 12

3.2 Time to Ruin . . . 12

3.3 Convergence to Cramér-Lundberg Model . . . 15

3.4 Simulation . . . 16

3.4.1 Code . . . 16

3.4.2 Results . . . 17

4 Model with linear and stochastic premium 19

4.1 Mathematical Model . . . 19

4.2 Auxiliary Model . . . 19

4.3 Convergence to Original Model . . . 23

4.4 Simulation . . . 25

4.4.1 Code . . . 25

4.4.2 Results . . . 26

5 Model with Markovian States 28

5.1 Mathematical Model . . . 28

5.2 Ruin Probability . . . 29

5.3 Simulation . . . 30

Table of contents

Title
Analysis of the Expected Time to Ruin in Several Insurance Models

5.3.1 Code 31
5.3.2 Results 32

6 Conclusion and discussion 38

7 Appendix 40

1 Introduction

People are exposed to various risks and uncertainties on an everyday basis. Whether it involves your health or your home, accidents can happen at any time. To protect themselves against such eventualities, people have come up with the concept of insurance.
Insurance in some forms is as old as historical society. Already in prehistoric times, people would make agreements of mutual aid with their families and neighbours, like helping each other with rebuilding a house when it gets destroyed. The first time that avoidance of risk has been transferred into monetary economy was approximately 4000 to 5000 years ago. Merchants would distribute their goods over several ships when sailing stormy seas and rough rivers to minimize the loss of goods due to the capsizing of a single ship. People have been trying to protect themselves against risk ever since [4].

Nowadays, insurance is an indispensable part of our society. People can insure themselves against almost every eventuality. As insurance deals with uncertainties and risk, it is needless to mention that it can be modelled using mathematics. One of the founders of mathematical risk theory is the Swedish mathematician Filip Lundberg (1876-1965). His main work on risk theory was published in the early 1900's and was very much ahead of its time, as at that time no general theory on stochastic processes existed. One of the only mathematicians that could decipher Lundberg's notes was another Swedish mathematician, named Harald Cramér. He republished Lundberg's work in 1930, introducing the Cramér-Lundberg model, also known as the classical (Poisson) risk model. This was the first model to describe insurer's cash flows using mathematics and is still used a lot nowadays. Many studies have taken this model as a basis and investigated variations and its properties, such as the probability that a company will go bankrupt.

As the title already suggests, in this report we will analyze several insurance models and look for the time to ruin of each model. The time to ruin is defined as the time it takes for an insurance company to go bankrupt, given that bankruptcy will actually occur. Self-evidently, we will look at the aforementioned Cramér-Lundberg model, as this is arguably the most fundamental insurance model. Through literature research, an explicit expression for the time to ruin of this model will be found. Furthermore, variations of the Cramér-Lundberg model with different ways of modelling cash flows of an insurance company will be investigated. Also for these other models, which have not been investigated as extensively as the Cramér-Lundberg model, the aim is finding analytic expressions for the time to ruin using various techniques. The structure of the bachelor final project of Sabine Geurts [3] has been taken as a starting point for this project and will be referred to regularly throughout the report.

1.1 Outline of the report

There are four main chapters in this report. In each chapter we will look for the time to ruin, first by finding an expression analytically and then by running a stochastic simulation. The first model that will be looked at is the Cramér-Lundberg model itself. The model will be formulated in mathematical terms and the results on the time to ruin have been found using the book "Insurance Risk and Ruin" by David Dickson [2]. In the next chapter, Chapter 3, a model with stochastic premium is discussed. A model description is given and the time to ruin will be computed analytically. The third model is a combination of the first two models with both linear and stochastic premium and is treated in Chapter 4. In the last chapter, a model is introduced that has not been discussed in Sabine Geurts' final project. This model has stochastic premium, similar to the model in the third chapter. However, there is a Markovian twist to this model, as we implement an underlying Markov process with two states affecting the claim sizes and the rate at which claims are coming in. For this model, no analytic expression for the expected time to ruin has been found, but a stochastic simulation still allows us to observe how this model behaves. At the end of this report, there is a conclusion and discussion to discuss the results, argue the reliability and make suggestions for future research.

2 Cramér-Lundberg Model

When analyzing the Cramér-Lundberg model and finding the time to ruin, the book "Insurance Risk and Ruin" has been used [2]. In particular, some of the results from chapter 7 and 8 are discussed.

2.1 The Classical Risk Process

Let us start with mathematically defining the Cramér-Lundberg model. This model assumes the premium comes in at a linear rate while the claims come in stochastically. The model looks as follows:

$$X(t) = x + ct - S(t). \tag{2.1}$$

Here $X(t)$ represents the insurer's surplus at time t, x is defined as the initial capital and c is the rate at which the premium comes in. Self-evidently, x and c are assumed to be non-negative. $S(t)$ is defined as $S(t) := \sum_{i=1}^{N(t)} Y_i$. Here Y_i is the size of the i-th claim and $N(t)$ represents the number of claims that occurred in the time interval $[0, t]$. We will denote the cumulative distribution function of the individual claims Y_i by $F(u) = \mathbb{P}(Y_i \leq u)$ and the moments of the individual claims by $\mu_j = \mathbb{E}[Y_i^j]$. In the classical risk process, $\{N(t)\}_{t\geq 0}$ is assumed to be a Poisson process with parameter λ, causing $S(t)$ to be a so-called compound Poisson process.

In this report, the focus will mainly be on the time to ruin, T_x, which is defined as $T_x := inf\{t \geq 0 : X(t) < 0, X(0) = x\}$, with $T_x = \infty$ if $X(t) \geq 0$ for all $t \geq 0$. Ruin is defined as the first time that the insurer's capital, $X(t)$, is smaller than zero. In some cases ruin might not occur, causing the time to ruin to be infinite. As we have no interest in those cases, the focus of this project will be on the expected time to ruin for cases where ruin does occur: $T_{x,f} := T_x | T_x < \infty$. Before being able to find an expression for the expected value of $T_{x,f}$, two other functions need to be defined.

Definition 2.1: *Let T_x be the time to ruin and x the initial capital. The **ruin probability**, $\psi : \mathbb{R}^+ \to [0, 1]$, is defined as $\psi(x) := \mathbb{E}[\mathbb{1}(T_x < \infty)]$, with $\mathbb{1}(T_x < \infty)$ being the indicator function that equals 1 if $T_x < \infty$ and 0 otherwise.*

Definition 2.2: *Let T be the time to ruin and x the initial capital. Then the function $\psi_n : \mathbb{R}^+ \to \mathbb{R}^+$ is defined as $\psi_n(x) := \mathbb{E}[T_x^n \cdot \mathbb{1}(T_x < \infty)]$ for $n \in \mathbb{N}$.*

Hence, we have that $\psi_n(x) = \psi(x)$, when $n = 0$. In order to keep this model (and all of the other insurance models) interesting, we use a so-called *safety loading condition*. This condition makes sure that the general drift of the surplus is positive and looks as follows: $c > \lambda\mu_1$. In other words, we want that $c = (1 + \theta)\lambda\mu_1$, where $\theta > 0$ is called the *safety loading factor* and $\mu_1 = \mathbb{E}[Y_i]$.

2.2 Time to Ruin

As a consequence of definitions 2.1 and 2.2, we know that the n-th moment of the expected time of ruin is given by

$$\mathbb{E}[T_x^n | T_x < \infty] = \mathbb{E}[T_{x,f}^n] = \frac{\psi_n(x)}{\psi(x)}. \tag{2.2}$$

If we would be able to find an expression for $\psi_n(x)$, we can substitute this into equation (2.2) and find an expression for $\mathbb{E}[T_{x,f}]$. In order to do so, we first have to compute two Laplace transforms which we will need when finding such an expression for $\psi_n(x)$. For the first one, a new function needs to be introduced. This function is generally known as a Gerber-Shiu function and allows us to analyze and compute ruin-related quantities. The Gerber-Shiu function is defined as

$$\chi(x) = \mathbb{E}[a(X_x, Y_x)e^{-\delta T_x}\mathbb{1}(T_x < \infty)], \tag{2.3}$$

where $a(x, y)$ is a non-negative function, commonly known as the "penalty function", X_x is the capital right before ruin, Y_x is the deficit right after ruin, $\delta \geq 0$ can be seen as the parameter of a Laplace transform and $\mathbb{1}$ is the indicator function that we have seen before.

Next, we will find a so-called integro-differential equation of $\chi(x)$ by conditioning on the first claim and then differentiating this expression. When a claim comes in, this happens between time t and $t + dt$ with probability $\lambda e^{-\lambda t} dt$. When the claim has arrived, it has probability $f(u)du$ of being of some size between u and $u + du$. The size of the claim can either be smaller than $x + ct$, causing a jump in the surplus from $(x + ct)$ to $(x + ct - u)$, or it can be greater than $x + ct$, which will result into ruin. Combining all this and integrating over all possible times and claim sizes gives us the following expression for $\chi(x)$:

$$\begin{aligned}\chi(x) &= \int_0^\infty \lambda e^{-\lambda t} e^{-\delta t} \int_0^{x+ct} \chi(x + ct - u) f(u) du\, dt \\ &+ \int_0^\infty \lambda e^{-\lambda t} e^{-\delta t} \int_{x+ct}^\infty a(x + ct, u - x - ct) f(u) du\, dt.\end{aligned} \tag{2.4}$$

Before we start differentiating, we substitute $s = x + ct$ into equation (2.4). This gives the integral formula

$$\begin{aligned}\chi(x) =& \frac{\lambda}{c} \int_x^\infty e^{-(\lambda+\delta)\frac{s-x}{c}} \int_0^s \chi(s - u) f(u) du\, ds \\ &+ \frac{\lambda}{c} \int_x^\infty \lambda e^{-(\lambda+\delta)\frac{s-x}{c}} \int_s^\infty a(s, u - s) f(u) du\, ds.\end{aligned}$$

When differentiating this expression and applying the Leibniz integral rule, we end up with the expression

$$\begin{aligned}\frac{d}{dx}\chi(x) =& \frac{\lambda(\lambda+\delta)}{c^2} \int_x^\infty e^{-(\lambda+\delta)\frac{s-x}{c}} \int_0^s \chi(s - u) f(u) du\, ds - \frac{\lambda}{c} \int_0^x \chi(x - u) f(u) du \\ &+ \frac{\lambda(\lambda+\delta)}{c^2} \int_x^\infty \lambda e^{-(\lambda+\delta)\frac{s-x}{c}} \int_s^\infty a(s, u - s) f(u) du\, ds - \frac{\lambda}{c} \int_x^\infty a(x, u - x) f(u) du\end{aligned}$$

$$\begin{aligned}\frac{d}{dx}\chi(x) - \frac{\lambda+\delta}{c} &\left(\frac{\lambda}{c} \int_x^\infty e^{-(\lambda+\delta)\frac{s-x}{c}} \int_0^s \chi(s - u) f(u) du\, ds + \frac{\lambda}{c} \int_x^\infty \lambda e^{-(\lambda+\delta)\frac{s-x}{c}} \int_s^\infty a(s, u - s) f(u) du\, ds \right) \\ &+ \frac{\lambda}{c} \int_0^x \chi(x - u) f(u) du + \frac{\lambda}{c} \int_x^\infty a(x, u - x) f(u) du = 0.\end{aligned}$$

We can rewrite in the following way:

$$\frac{d}{dx}\chi(x) - \frac{\lambda+\delta}{c}\chi(x) + \frac{\lambda}{c} \int_0^x \chi(x - u) f(u) du + \frac{\lambda}{c}\alpha(x) = 0, \tag{2.5}$$

where

$$\alpha(x) = \int_0^\infty a(x, y) f(y + x) dy.$$

and with $y = u - x$. The next step is to take the Laplace transform of equation (2.5). For this, let us define $\chi^*(s) := \int_0^\infty e^{sx} \chi(x) dx$. Also, recall two properties of the Laplace transform that we will need:

- Laplace transform of a derivative: $\mathcal{L}(\frac{d}{dx} f(x)) = s\mathcal{L}(f(x)) - f(0)$
- Laplace transform of a convolution: $\mathcal{L}\left(\int_0^x f(u) g(x - u) du\right) = \mathcal{L}(f(x)) \cdot \mathcal{L}(g(x))$

Applying these properties to equation (2.5) gives the following expression:

$$s\chi^*(s) - \chi(0) - \frac{\lambda+\delta}{c}\chi^*(s) + \frac{\lambda}{c} f^*(s)\chi^*(s) + \frac{\lambda}{c}\alpha^*(s) = 0.$$

Rewriting this equation results into the following expression, which will be our starting point for finding an expression for $\psi_n(x)$:

$$\chi^*(s) = \frac{c\chi(0) - \lambda\alpha^*(s)}{cs - \lambda - \delta + \lambda f^*(s)}. \tag{2.6}$$

Before we introduce the theorem that will state the expression for $\psi_n(x)$, we have to repeat these steps but with the survival probability $\phi(x)$, which can be seen as $1 - \psi(x)$. Again, we will come up with an integral formula for the survival probability by conditioning on the first claim:

$$\phi(x) = \int_0^\infty \lambda e^{-\lambda t} \int_0^{x+ct} \phi(x + ct - u) f(u) du\, dt. \tag{2.7}$$

Substituting $s = x + ct$ and differentiating gives the following result:

$$\frac{d}{dx}\phi(x) = \frac{\lambda^2}{c^2} \int_0^\infty e^{-\lambda \frac{s-x}{c}} \int_0^s \phi(s-u) f(u) du\, dt - \frac{\lambda}{c} \int_0^x \phi(x-u) f(u) du$$

$$\frac{d}{dx}\phi(x) - \frac{\lambda}{c}\phi(x) + \frac{\lambda}{c}\int_0^x \phi(x-u) f(u) du = 0. \tag{2.8}$$

Taking the Laplace transform of equation (2.8) gives the expression

$$s\phi^*(s) - \phi(0) - \frac{\lambda}{c}\phi^*(s) + \frac{\lambda}{c} f^*(s)\phi^*(s) = 0.$$

Rewriting this gives the expression for the Laplace transform of the survival probability

$$\phi^*(s) = \frac{c\phi(0)}{cs - \lambda + \lambda f^*(s)}. \tag{2.9}$$

Now we have found both equations (2.6) and (2.9), we can introduce the theorem that states the recursion formula for $\psi_n(x)$:

Theorem 2.1: *$\psi_n(x)$ can be expressed in terms of $\psi_{n-1}(\cdot)$ with the following recursion formula:*

$$\psi_n(x) = \frac{n}{c\phi(0)} \left(\phi(x) \int_0^\infty \psi_{n-1}(y) dy - \int_0^x \psi_{n-1}(u)\phi(x-u) du \right). \tag{2.10}$$

Proof. In this proof we will make use of the fact that the Gerber-Shiu function (2.3) is just the Laplace transform of T_x when $a(x, y) = 1$. Hence:

$$\chi(x) = \int_0^\infty e^{-\delta t}\omega(x, t) dt,$$

where $\omega(x, t)$ represents the density of T_x, such that

$$\mathbb{P}(T_x \le t) = \int_0^t \omega(x, s) ds.$$

We will start by rewriting equation (2.6) in the following way:

$$(cs - \lambda + \lambda f^*(s))\chi^*(s) - \delta\chi^*(s) = c\chi(0) + \lambda\alpha^*(s). \tag{2.11}$$

The next step is to differentiate this equation n times with respect to δ. As $\alpha^*(s)$ is independent of δ, this expression will disappear when differentiating. When considering the term $\delta\chi^*(s)$, it is straightforward to show by induction that:

$$\frac{d^n}{d\delta^n}(\delta\chi^*(s)) = n\frac{d^{n-1}}{d\delta^{n-1}}\chi^*(s) + \delta\frac{d^n}{d\delta^n}\chi^*(s).$$

Furthermore, for convenience, we define:

$$k_n(s) := \frac{d^n}{d\delta^n}\chi^*(s)\Big|_{\delta=0},$$

which actually represents the Laplace transform of $(-1)^n\psi_n(x)$:

$$\chi^*(s) = \int_0^\infty e^{-sx}\chi(x)dx = \int_0^\infty e^{-sx}\int_0^\infty e^{-\delta t}\omega(x,t)dt\,dx$$

$$\frac{d^n}{d\delta^n}\chi^*(s) = \int_0^\infty e^{-sx}(-1)^n\int_0^\infty t^n e^{-\delta t}\omega(x,t)dt\,dx.$$

When we evaluate this expresssion at $\delta = 0$, we get:

$$\frac{d^n}{d\delta^n}\chi^*(s)\Big|_{\delta=0} = \int_0^\infty e^{-sx}(-1)^n\int_0^\infty t^n\omega(x,t)dt\,dx,$$

and hence

$$k_n(s) = \frac{d^n}{d\delta^n}\chi^*(s)\Big|_{\delta=0} = \int_0^\infty e^{-sx}(-1)^n\psi_n(x)dx$$

as we know that $\psi_n(x) = \mathbb{E}[T_x^n \cdot \mathbb{1}(T_x < \infty)] = \int_0^\infty t^n\omega(x,t)dt$. Now we can differentiate equation (2.11) n times and substitute $\delta = 0$. This gives the following equation:

$$(cs - \lambda + \lambda f^*(s))k_n(s) - nk_{n-1}(s) = (-1)^n c\psi_n(0).$$

Rewriting this gives us the following equation:

$$k_n(s) = \frac{(-1)^n c\psi_n(0) + nk_{n-1}(s)}{cs - \lambda + \lambda f^*(s)}. \tag{2.12}$$

When we recall the Laplace transform of the survival probability, equation (2.9), we can combine that equation with equation (2.12) to get the following expression:

$$k_n(s) = \frac{\phi^*(s)}{c\phi(0)}\big((-1)^n c\psi_n(0) + nk_{n-1}(s)\big). \tag{2.13}$$

The next step is to take the inverse Laplace transform of this equation. We will rewrite the equation first to make things easier:

$$k_n(s) = (-1)^n\psi_n^*(s) = (-1)^n\frac{\psi_n(0)}{\phi(0)}\phi^*(s) + (-1)^{n-1}\frac{n}{c\phi(0)}\phi^*(s)\psi_{n-1}^*(s).$$

Now taking the inverse Laplace transform yields:

$$\psi_n(x) = \frac{\psi_n(0)}{\phi(0)}\phi(x) - \frac{n}{c\phi(0)}\int_0^x \psi_n(u)\phi(x-u)du. \tag{2.14}$$

Now the last thing we need to do to complete the proof is to find an expression for $\psi_n(0)$. We can do this by rewriting equation (2.12) in the following way:

$$(cs - \lambda + \lambda f^*(s))k_n(s) = nk_{n-1}(s) + (-1)^n c\psi_n(0). \tag{2.15}$$

We know, as $f^*(0) = \int_0^\infty f(u)du = 1$, that the left-hand side of equation (2.15) is 0 when $s = 0$. Therefore, the right-hand side must also be equal to 0:

$$\psi_n(0) = \frac{n}{c}(-1)^{n-1}k_{n-1}(0) = \frac{n}{c}\int_0^\infty \psi_{n-1}(y)dy.$$

Substituting this in equation (2.14), gives us the following recursion formula for $\psi_n(x)$:

$$\psi_n(x) = \frac{n}{c\phi(0)}\left(\phi(x)\int_0^\infty \psi_{n-1}(y)dy - \int_0^x \psi_{n-1}(u)\phi(x-u)du\right). \tag{2.16}$$

□

2.3 Exponential Claims

Now that we have found an expression for $\psi_n(x)$, the next step is to find an expression for the expected time to ruin, using this result. In order to do this, we first need to specify the distribution of the individual claims. It is common to use an exponential distribution for the claim sizes, meaning $f(u) = \alpha e^{-\alpha u}$. When the distribution of the claim sizes is exponential, the book "Insurance Risk and Ruin" has derived a useful result for the ruin probability, namely that $\psi(x) = \psi(0)e^{-Rx}$, where $\psi(0) = \frac{\lambda}{\alpha c}$ and $R = \alpha - \frac{\lambda}{c}$. Taking this result and equation (2.16) into account, we can now find an expression for the expected time to ruin:

Theorem 2.2: *Let $f(u) = \alpha e^{-\alpha u}$ such that $\psi(x) = \frac{\lambda}{\alpha c} e^{-(\alpha - \frac{\lambda}{c})x}$. Then we have that:*

$$\mathbb{E}[T_{x,f}] = \frac{c + \lambda x}{c(\alpha c - \lambda)}. \tag{2.17}$$

Proof. We start by substituting $n = 1$ into equation (2.16). This gives:

$$\psi_1(x) = \frac{1}{c\phi(0)} \left(\phi(x) \int_0^\infty \psi(x)dx - \int_0^x \psi(u)\phi(x-u)du \right). \tag{2.18}$$

We know that:

$$\int_0^\infty \psi(x)dx = \left[-\frac{\psi(0)}{R} e^{-Rx} \right]_0^\infty = \frac{\psi(0)}{R}.$$

Furthermore, we have that:

$$\int_0^x \psi(u)\phi(x-u)du = \int_0^x \psi(u)(1-\psi(x-u))du = \int_0^x \psi(0)e^{-Ru}(1-\psi(0)e^{-R(x-u)})du$$

$$= \psi(0)\int_0^x e^{-Ru} - \psi(0)e^{-Rx}du = \frac{\psi(0)}{R}(1-e^{-Rx}) - \psi(0)^2 e^{-Rx}x.$$

Substituting these two expressions into equations (2.18) gives the result

$$\psi_1(x) = \frac{\psi(0)}{Rc(1-\psi(0))} \left(1 - \psi(x) - 1 + e^{-Rx} + \psi(0)e^{-Rx}Rx \right).$$

When we divide this equation by $\psi(x) = \psi(0)e^{-Rx}$, we get an expression for $\mathbb{E}[T_{x,f}]$:

$$\mathbb{E}[T_{x,f}] = \frac{\psi(0)}{Rc(1-\psi(0))} \left(-1 + \frac{1}{\psi(0)} + Rx \right).$$

Substituting $\psi(0) = \frac{\lambda}{\alpha c}$ and $R = \alpha - \frac{\lambda}{c}$ will, after some algebra, give the following equation eventually:

$$\mathbb{E}[T_{x,f}] = \frac{c + \lambda x}{c(\alpha c - \lambda)}. \tag{2.19}$$

□

Now that we have obtained an expression for the time to ruin of the Cramér-Lundberg model, we will try and do the same in the next chapters for the other models. However, first we will use a simulation to check and visualize our findings.

2.4 Simulation

Not only do we want to come up with an exact expression for the expected time to ruin, we also would like to see some numerical results by simulating the classical risk process. In that way we can visualize how certain parameters influence the model and whether the mathematical expression that we found actually matches the numerical results. Hence, at the end of each chapter we will simulate the corresponding model and show the results.

2.4.1 Code

The simulation is written using Python as this is a versatile and easy to use programming language. The code of this first model has been written in such a way that without much modification it can be used for the other models as well. In total, four classes have been used for this simulation:

1. **Main Simulation:** This is where the main model is written. The algorithm that has been used is explained in the pseudo-code of algorithm 1.
2. **Event:** In this class the types of events (claim or premium) and their attributes are defined.
3. **FES:** This class is used for the scheduling of the future events.
4. **Simulate Results:** As the name suggests, this class is used for visualizing the results.

The main simulation looks as follows:

Algorithm 1 Cramér-Lundberg Simulation

Initialize all parameters;
Initialize FES() and SimulateResults();
Initialize time and initial capital;
Schedule first claim;
while *Capital* ≥ 0 & $T < T_{max}$ **do**
    Jump to the next event;
    Update the linear capital increase;
    Store the current time and capital;
    if *Next event is claim* **then**
        Subtract the claim amount from the capital;
        Store the time and capital;
        Schedule the next claim;
    end
end
return SimulateResults() and time;

We have to somehow define a time limit, such that if after this time no ruin has occurred yet, we assume it never will. That value is what is called T_{max} in the algorithm. We can choose it by trying multiple values. By letting the algorithm returning the time, we can check whether this time is actually smaller than T_{max}. If it is, we know that ruin occurred and otherwise we discard that run. By running many times and only considering the results where the time is smaller than T_{max}, we can get accurate estimations for the time to ruin and the ruin probability. Obviously, the more runs we do, the more accurate the outcome will be. To measure the accuracy of the results, we will use 95%-confidence intervals. Algorithm 2 shows how the time to ruin, ruin probability and their confidence intervals are computed:

Algorithm 2 Compute time to ruin and ruin probability

Initialize number of runs N;
Create empty results vector;
for *i in range(N)* **do**
  Run the Cramér-Lundberg Simulation;
  if *time* $\leq T_{max}$ **then**
    Add the time to results vector;
  end
end
Time to Ruin = $\hat{T}$ = mean(results);
Ruin probability = $\hat{p}$ = length(results) / N;
$CI_{\text{Time To Ruin}} = \hat{T} \pm 1.96 \cdot \frac{\text{standard deviation}}{\text{length(results)}}$
$CI_{\text{Ruin Probability}} = \hat{p} \pm \frac{1.96 \cdot \sqrt{\hat{p}(1-\hat{p}}}{\sqrt{N}}$

With the use of these two algorithms, we can now compute numerical values for the time to ruin and ruin probability and compute their accuracy by using confidence intervals.

2.4.2 Results

The first thing we would like to do is to observe whether a single run would look like we would expect; with a positive linear rate and a downward jump every now and then. For this we first need to specify the parameters. For this single run, we will take $x = 5$, $c = 2$, $\lambda = 1$ and exponentially distributed claims with $\mathbb{E}[Y_i] = 1.5$. Hence the safety loading condition is satisfied, as $2 > 1 \cdot 1.5$. We get the following plot:

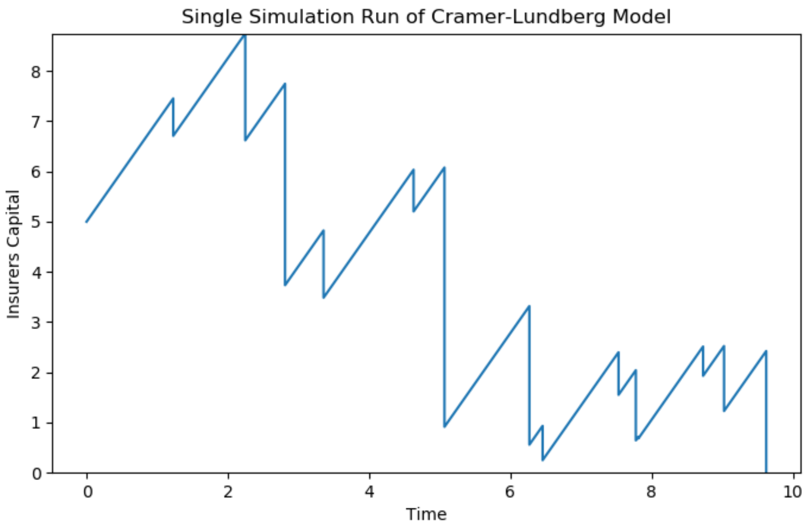


Figure 2.1: Plot of a single run of the Cramér-Lundberg simulation.

This plot clearly matches our expectations. We can see that in this specific run ruin occurred at around $t = 9.5$. Hence, in this case the algorithm would return $t = 9.5$.

Now that we know the simulation does what we would like, we can start computing numerical results by simulating a greater amount of runs. For this, we need to choose a value for T_{max}. If we take this value too low, some of the runs where ruin was still going to occur will be discarded. However, taking this value too large will cause the simulation to do many unnecessary computations. The effect of the value of T_{max} on the outcome is shown in the figure below, where the red line represents the theoretical value for the expected time to ruin using this particular parameter set:

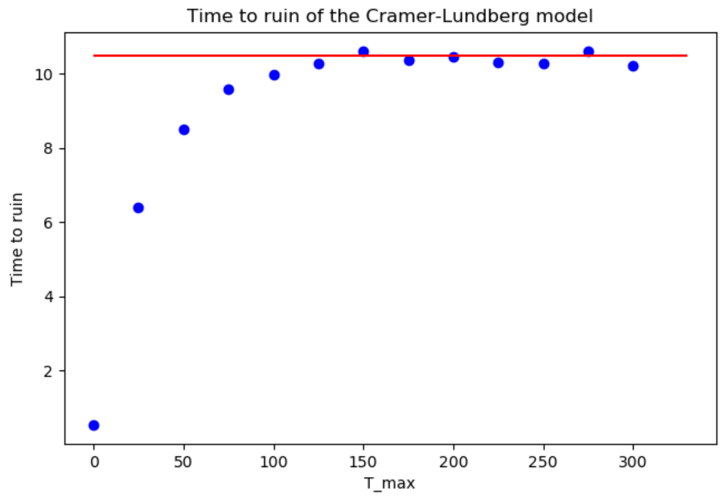


Figure 2.2: How the value for T_{max} affects the outcome of the numerical results.

In this plot, we can see that around $T_{max} = 150$, the value of the expected time to ruin does not change anymore. Hence for this set of parameters, we know that $T_{max} = 150$ is a reasonable value for T_{max} when a trade-off has to be made between simulation accuracy and simulation speed. However, when parameters such as the initial capital x or number of claims per time unit λ would change, the optimal value of T_{max} will change as well. Hence, for each different set of parameters we have to check whether the value of T_{max} is not chosen too small or too large.

Obviously, we want the numerical results to be as accurate as possible so we want to do as many runs as possible. However, due to time restrictions we cannot take the number of runs too large. The table below shows how the number of runs affects the accuracy of the simulation results for the expected time to ruin and ruin probability, for the same set of parameters as in Figure 2.1.

Table 2.1: The effect of the number of runs on the size of the confidence intervals.

Number of runs	$N = 1000$	$N = 5000$	$N = 10000$
Time to Ruin	$(9.217, 12.295)$	$(9.944, 11.549)$	$(9.926, 10.963)$
Ruin Probability	$(0.305, 0.363)$	$(0.312, 0.338)$	$(0.321, 0.339)$

As expected, the confidence intervals become smaller when the number of runs increases. For $N = 10000$ we get sufficient small confidence intervals with an acceptable simulation time. Therefore, we will keep doing 10000 runs for the remainder of this section.

Next, the goal is to simulate the time to ruin as a function of the initial capital. With a for-loop, we will loop over different values for the initial capital, x, and plot the results. For now, the same parameters as in Figure 2.1 will be used. In the same plot, the line $\mathbb{E}[T_{x,f}] = \frac{c+\lambda x}{c(\alpha c-\lambda)}$ is plotted to see whether the theoretical analysis matches the numerical results. This plot looks as follows:

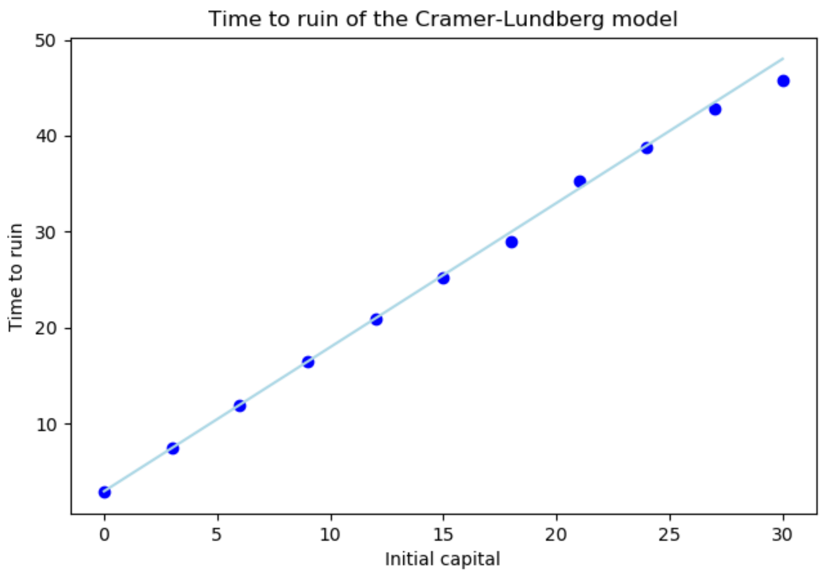


Figure 2.3: Plot of the theoretical and numerical results.

Up and until $x = 15$ the numerical results lie exactly on the line. After that, the numerical results start to slightly deviate from the line. This is a result of the fact that a higher initial capital causes the ruin probability to be smaller. As the runs where no ruin occurs are useless when computing the time to ruin, there will be fewer runs left to work with as the ruin probability increases, causing the accuracy of the simulation to decrease. In the table below, some of the numerical results of the ruin probabilities for different initial capitals are shown:

Table 2.2: The effect of the initial capital on the ruin probability.

Initial capital	$x = 0$	$x = 9$	$x = 15$	$x = 30$
Ruin Probability	0.751	0.166	0.061	0.005

This means that we have to keep track of the ruin probability in order to keep the results for the time to ruin reasonably accurate. Hence, for parameter sets with different ruin probabilities we might need a different amount of runs.

The last thing we will do in this chapter is to observe how the time to ruin is affected by choosing different sets of parameters. The results of these different parameter sets are plotted in the figure below, where the legend displays what parameters are being used.

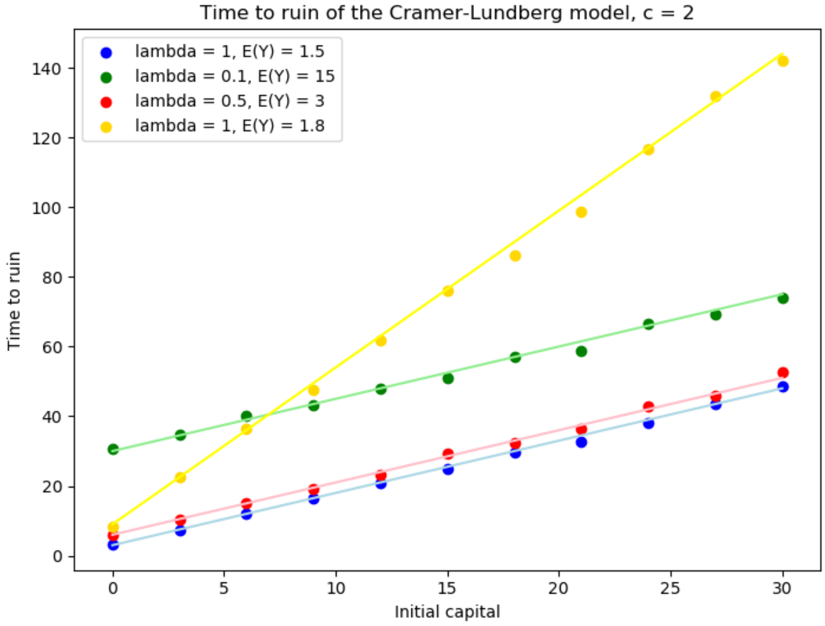


Figure 2.4: Plot of the theoretical and numerical results of multiple parameter sets.

In the plot the premium rate has been kept constant at $c = 2$, while the expected claim size, $\mathbb{E}[Y_i] = \frac{1}{\alpha}$, and the compound Poisson parameter, λ, are varied. What stands out is that when the safety loading is not changed, which is the case with the blue, red and green line (as $1 \cdot 1.5 = 0.1 \cdot 15 = 0.5 \cdot 3$), the lines are parallel. It can be observed that when the claims come in less frequently but with a higher average amount, the time to ruin is higher. We can explain this by considering what happens at $t = 0$. Due to a low value for λ, it will on average take longer before the first claim comes in. Regardless of the claim size of this claim, ruin can only occur as soon as a claim comes in. So if it takes longer for the first claim to arrive, it will automatically take longer for ruin to occur. When the safety loading is decreased, which is done with the yellow line relative to the blue line, the slope turns out to be much steeper. When an insurance company has a higher average profit (which we can define as income minus expenses), their capital will go to infinity more quickly. Hence, if ruin were to occur, it would happen relatively quick, as the probability is very low that ruin would still occur when a company's capital is big. When the average profit is relatively low, the company's capital will rise less quickly. This causes the probability that ruin occurs at a later point in time to be higher.

3 Model with stochastic premium

In this chapter we will discuss a different model, one that has first been introduced in an article by A.V. Bolkov, called *"The Cramér-Lundberg model with stochastic premium process"* [1]. We will again look for an equation for the expected time to ruin. Furthermore, we will show that this model actually converges to the Cramér-Lundberg model for a certain choice of parameters.

3.1 The Mathematical Model

This model is quite similar to the Cramér-Lundberg model. Again, we consider $X(t)$, denoting the insurer's surplus at time t. We also begin with a (non-negative) initial capital of x at time $t = 0$ and have a compound Poisson process describing the arrival of the claims. However, instead of linear premium, we now also have a compound Poisson process for the premiums, which is independent of the claim process. So we get a mathematical model which looks as follows:

$$X(t) = x + P(t) - S(t), \tag{3.1}$$

where $P(t) := \sum_{j=1}^{N^+(t)} Z_j$ and $S(t) := \sum_{i=1}^{N^-(t)} Y_i$. Here, $\{N^+(t)\}_{t\geq 0}$ with intensity λ^+ represents the Poisson process that describes the number of premiums that have arrived within the time interval $[0, t]$, causing $P(t)$ to be a compound Poisson process. Now Z_j is the size of the j-th premium, with $j \in \{1, 2, 3, ...\}$, and has cumulative distribution function $G(v) = \mathbb{P}(Z_j \leq v)$.
$\{N^-(t)\}_{t>0}$ with intensity λ^- represents the Poisson process describing the number of claims within the time interval $[0, t]$ and is independent of $N^+(t)$. Hence, $S(t)$ is also a compound Poisson process. Again, Y_i is the size of the i-th claim, with $i \in \{1, 2, 3, ...\}$, and has cumulative distribution function $F(u) = \mathbb{P}(Y_i \leq u)$. Obviously, both Y_i and Z_j are assumed to be non-negative. The safety loading condition for this model, which will mainly be used in the simulation, looks as follows: $\lambda^+ \cdot \mathbb{E}[Z_j] > \lambda^- \cdot \mathbb{E}[Y_i]$.

3.2 Time to Ruin

Again, we will look for an expression for the expected time to ruin. Obviously, it still holds that $\mathbb{E}[T_{x,f}] = \frac{\psi_1(x)}{\psi(x)}$ and we will again use this in our search for the expected time to ruin. Furthermore, we will use a result that has been found in the bachelor final project of Sabine Geurts , which states that the ruin probability of this model is equal to $\psi(x) = C \cdot e^{-Rx}$, where $C = \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^++\lambda^-)}$ and $R = \frac{\lambda^+\alpha-\lambda^-\beta}{\lambda^++\lambda^-}$ [3]. We start by considering $\psi_1(x)$ and finding an integral equation by conditioning on the first event:

Theorem 3.1. *$\psi_1(x)$ satisfies the following integral equation:*

$$(\lambda^+ + \lambda^-)\psi_1(x) = \psi(x) + \lambda^- \int_0^x \psi_1(x-u)f(u)du + \lambda^+ \int_0^\infty \psi_1(x+v)g(v)dv. \tag{3.2}$$

Proof. For this proof we will consider what will happen first after time $t = 0$. Of course, for a certain amount of time, no claims or premiums will come in. The expected amount of time that no claims or premiums will occur, is $\frac{1}{\lambda^++\lambda^-}$. However, this time will only contribute to $\psi_1(x)$ when ruin actually occurs. Otherwise the indicator function, $\mathbb{1}(T < \infty)$, will cause $\psi_1(x)$ to be zero. The probability that ruin eventually occurs is $\psi(x)$. Then at some point, either a premium comes in, with probability $\frac{\lambda^+}{\lambda^++\lambda^-}$, or a claim comes in with probability $\frac{\lambda^-}{\lambda^++\lambda^-}$. When a premium comes in, this premium has probability $g(v)dv$ to be of a size somewhere between v and $v + dv$. The premium causes a jump in the surplus to level $(x + v)$. The same can be done for a claim: with probability $f(u)du$ a claim of a size between u and $u + du$ comes in, causing a jump in the surplus to level $(x - u)$. Clearly, a premium can be of any size between 0 and ∞. However, a claim can be of size at most x, as it will result into ruin otherwise. Combining these results gives us the following equation:

$$\psi_1(x) = \frac{1}{\lambda^+ + \lambda^-}\psi(x) + \frac{\lambda^-}{\lambda^+ + \lambda^-}\int_0^x \psi_1(x-u)f(u)du + \frac{\lambda^+}{\lambda^+ + \lambda^-}\int_0^\infty \psi_1(x+v)g(v)dv,$$

which obviously results in the integral equation (3.2). □

The next step is to find an expression for $\psi_1(x)$, as this will allow us to get a closed-form expression for the expected time to ruin. Again, we will make the assumption that the claim sizes are exponentially distributed, as we did with the Cramér-Lundberg model. Furthermore, we also assume that the size of the premium is exponentially distributed. Then we can prove the following theorem:

Theorem 3.2: *Let $F(u) = 1 - e^{-\alpha u}$, $G(v) = 1 - e^{-\beta v}$ and $\alpha, \beta > 0$. Then $\psi_1(x)$ equals:*

$$\psi_1(x) = \frac{\lambda^+(\alpha+\beta)}{(\lambda^+ + \lambda^-)(\lambda^+\alpha - \lambda^-\beta)}\psi(x) + x \cdot \frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+ + \lambda^-)^2(\lambda^+\alpha - \lambda^-\beta)}\psi(x)$$

Proof. We will use theorem 3.1 and rewrite equation (3.2) in the following way:

$$(\lambda^+ + \lambda^-)\psi_1(x) - \psi(x) = \lambda^- \int_0^x \psi_1(x-u)\alpha e^{-\alpha u}du + \lambda^+ \int_0^\infty \psi_1(x+v)\beta e^{-\beta v}dv. \tag{3.3}$$

We are going to differentiate this equation on both sides. In order to do this, we will introduce two new variables: $x_u := x - u$ and $x_v := x + v$. Substituting this in equation (3.3) yields:

$$(\lambda^+ + \lambda^-)\psi_1(x) - \psi(x) = \lambda^- \int_0^x \psi_1(x_u)\alpha e^{-\alpha(x-x_u)}dx_u + \lambda^+ \int_x^\infty \psi_1(x_v)\beta e^{-\beta(x_v-x)}dx_v.$$

We can differentiate this equation using the Leibniz integral rule. This gives the following derivatives:

$$\begin{aligned}\frac{d}{dx}\int_0^x \psi_1(x_u)\alpha e^{-\alpha(x-x_u)}dx_u &= \psi_1(x)\alpha - \alpha\int_0^x \psi_1(x_u)\alpha e^{-\alpha(x-x_u)}dx_u \\ &= \psi_1(x)\alpha - \alpha\int_0^x \psi_1(x-u)\alpha e^{-\alpha u}du\end{aligned}$$

The same can be done for the other integral:

$$\begin{aligned}\frac{d}{dx}\int_x^\infty \psi_1(x_v)\beta e^{-\beta(x_v-x)}dx_v &= -\psi_1(x)\beta + \beta\int_x^\infty \psi_1(x_v)\beta e^{-\beta(x_v-x)}dx_v \\ &= -\psi_1(x)\beta + \beta\int_0^\infty \psi_1(x+v)\beta e^{-\beta v}dv\end{aligned}$$

Therefore, if we differentiate equation (3.3), we get:

$$(\lambda^+ + \lambda^-)\psi_1'(x) - \psi'(x) = -\lambda^+\beta\psi_1(x) + \beta\lambda^+\int_0^\infty \psi_1(x+v)\beta e^{-\beta v}dv + \lambda^-\alpha\psi_1(x) - \alpha\lambda^-\int_0^x \psi_1(x-u)\alpha e^{-\alpha u}du,$$

which we can rewrite as

$$\begin{aligned}(\lambda^+ + \lambda^-)\psi_1'(x) + (\lambda^+\beta - \lambda^-\alpha)\psi_1(x) - \psi'(x) = \beta\lambda^+ &\int_0^\infty \psi_1(x+v)\beta e^{-\beta v}dv \\ - \alpha\lambda^- &\int_0^x \psi_1(x-u)\alpha e^{-\alpha u}du.\end{aligned} \tag{3.4}$$

We can differentiate this equation again, using the same technique as we used with equation (3.3). Doing this will eventually result in the following equation:

$$(\lambda^+ + \lambda^-)\psi_1''(x) + (\lambda^+\beta - \lambda^-\alpha)\psi_1'(x) + (\alpha^2\lambda^- + \beta^2\lambda^+)\psi_1(x) - \psi''(x) = \beta^2\lambda^+ \int_0^\infty \psi_1(x+v)\beta e^{-\beta v}dv + \alpha^2\lambda^- \int_0^x \psi_1(x-u)\alpha e^{-\alpha u}du \tag{3.5}$$

Now let us define $I_\alpha := \int_0^x \psi_1(x-u)\alpha e^{-\alpha u}du$ and $I_\beta := \int_0^\infty \psi_1(x+v)\beta e^{-\beta v}dv$. Then we can see that the right hand sides of equations (3.3), (3.4) and (3.5) look as follows:

3.3: $\lambda^+ I_\beta + \lambda^- I_\alpha$

3.4: $\beta\lambda^+ I_\beta - \alpha\lambda^- I_\alpha$

3.5: $\beta^2\lambda^+ I_\beta + \alpha^2\lambda^- I_\alpha$

It then becomes clear that we can write equation (3.5) as a linear combination of equations (3.3) and (3.4), namely: $(3.5) = \alpha\beta \cdot (3.3) + (\beta - \alpha) \cdot (3.4)$. We can do the same with the left hand side of the three equations. After some algebra, this eventually gives us the following equation:

$$\psi_1''(x)(\lambda^+ + \lambda^-) + \psi_1'(x)(\lambda^+\alpha - \lambda^-\beta) + \alpha\beta\psi(x) + (\beta - \alpha)\psi'(x) - \psi''(x) = 0 \tag{3.6}$$

Now consider the second half of the left hand side of equation (3.6): $\alpha\beta\psi(x) + (\beta-\alpha)\psi'(x) - \psi''(x)$. We know that $\psi(x) = \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^+ + \lambda^-)} exp(\frac{\lambda^-\beta - \lambda^+\alpha}{\lambda^+ + \lambda^-}x)$. Using this, we can express the derivatives $\psi'(x)$ and $\psi''(x)$ in terms of $\psi(x)$: $\psi'(x) = \frac{\lambda^-\beta - \lambda^+\alpha}{\lambda^+ + \lambda^-} \cdot \psi(x)$ and $\psi''(x) = \left(\frac{\lambda^-\beta - \lambda^+\alpha}{\lambda^+ + \lambda^-}\right)^2 \cdot \psi(x)$. When we substitute these two expressions in the second half of the left hand side of equation (3.6), it turns out that all the terms in equation (3.6) that contain $\psi(x)$ can be rewritten as $\frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+ + \lambda^-)^2}$. All in all, this gives us the following non-homogeneous differential equation:

$$\psi_1''(x) - P\psi_1'(x) = -\frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+ + \lambda^-)^3} \cdot \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^+ + \lambda^-)} \cdot e^{Px}, \tag{3.7}$$

where $P = \frac{\lambda^-\beta - \lambda^+\alpha}{\lambda^+ + \lambda^-}$.
As the solutions to the auxiliary equation from the homogeneous equation, $m^2 - Pm = 0$, are P and 0, we know that the complementerary solution to the homogeneous part of the differentail equation will look like this: $\psi_1(x)_H = C_1 + C_2 e^{Px}$. The non-homogeneous solution will look like this: $\psi_1(x)_N = Axe^{Px}$, as P is a solution of the homogeneous part. Next, we compute the derivative and the second derivative of the non-homogeneous solution: $\psi_1'(x)_N = Ae^{Px}(1 + Px)$ and $\psi_1''(x)_N = APe^{Px}(2 + Px)$. Substituting these two expressions in equation (3.7) gives us the following expression for A: $A = \frac{\lambda^+(\lambda^-)^2(\alpha+\beta)^3}{\alpha(\lambda^+\alpha - \lambda^-\beta)(\lambda^+ + \lambda^-)^3}$. Adding the homogeneous and non-homogeneous part together, gives the following:

$$\psi_1(x) = C_1 + C_2 e^{Px} + x \cdot \frac{\lambda^+(\lambda^-)^2(\alpha+\beta)^3}{\alpha(\lambda^+\alpha - \lambda^-\beta)(\lambda^+ + \lambda^-)^3} e^{Px},$$

which can be written as:

$$\psi_1(x) = C_1 + C_2 e^{Px} + x \cdot \frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+\alpha - \lambda^-\beta)(\lambda^+ + \lambda^-)^2} \psi(x). \tag{3.8}$$

Now we are only left with finding expressions for the constants C_1 and C_2. First of all, we know that when the initial capital goes to infinity, ruin will never occur. Therefore, as a consequence of the indicator function in the definition of $\psi_1(x)$, we know that $\lim_{x\to\infty} \psi_1(x) = 0$. As both $\psi(x)$ and e^{Px}

will go to 0 (for $\lambda^+\alpha > \lambda^-\beta$) when $x \to \infty$, we know that $\lim_{x\to\infty} \psi_1(x) = C_1 = 0$. Now the last thing to do is to find C_2. For that, we will look at $\psi_1(0)$:

$$(\lambda^+ + \lambda^-)\psi_1(0) = \psi(0) + \lambda^+ \int_0^\infty \psi_1(v)\beta e^{-\beta v} dv$$

$$(\lambda^+ + \lambda^-)C_2 = \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^+ + \lambda^-)} + \lambda^+\beta \int_0^\infty \left(C_2 e^{Pv} + v \cdot \frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+\alpha - \lambda^-\beta)(\lambda^+ + \lambda^-)^2}\psi(v) \right) e^{-\beta v} dv$$

$$(\lambda^+ + \lambda^-)C_2 = \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^+ + \lambda^-)} + \lambda^+\beta C_2 \int_0^\infty e^{(-\beta+P)v} dv + \lambda^+\beta \frac{\lambda^+(\lambda^-)^2(\alpha+\beta)^3}{\alpha(\lambda^+\alpha - \lambda^-\beta)(\lambda^+ + \lambda^-)^3} \int_0^\infty v e^{(-\beta+P)v} dv$$

$$(\lambda^+ + \lambda^-)C_2 = \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^+ + \lambda^-)} + \frac{\beta(\lambda^- + \lambda +)}{\alpha+\beta} C_2 + \frac{(\lambda^-)^2(\alpha+\beta)\beta}{\alpha(\lambda^+\alpha - \lambda^-\beta)(\lambda^+ + \lambda^-)}$$

After some algebra, this gives the following expression for C_2:

$$C_2 = \frac{\lambda^+(\alpha+\beta)}{(\lambda^+ + \lambda^-)(\lambda^+\alpha - \lambda^-\beta)} \cdot \frac{(\alpha+\beta)\lambda^-}{\alpha(\lambda^+ + \lambda^-)}.$$

Note that we deliberately split the expression in two parts, such that it becomes obvious that when we substitute this expression in equation (3.8), we end up with the following expression for $\psi_1(x)$:

$$\psi_1(x) = \frac{\lambda^+(\alpha+\beta)}{(\lambda^+ + \lambda^-)(\lambda^+\alpha - \lambda^-\beta)}\psi(x) + x \cdot \frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+ + \lambda^-)^2(\lambda^+\alpha - \lambda^-\beta)}\psi(x)$$

□

Now we can use the fact that $\mathbb{E}[T_{x,f}] = \frac{\psi_1(x)}{\psi(x)}$ and we get the following corollary:

Corollary 3.2.1: The expected time to ruin $\mathbb{E}[T_{x,f}]$ is given by:

$$\mathbb{E}[T_{x,f}] = \frac{\lambda^+(\alpha+\beta)}{(\lambda^+ + \lambda^-)(\lambda^+\alpha - \lambda^-\beta)} + \frac{\lambda^+\lambda^-(\alpha+\beta)^2}{(\lambda^+ + \lambda^-)^2(\lambda^+\alpha - \lambda^-\beta)} \cdot x. \tag{3.9}$$

Now that we have found an expression for the time to ruin for both the Cramér-Lundberg model and the model with stochastic premium, we can see if there are any similarities and if they are related in some way.

3.3 Convergence to Cramér-Lundberg Model

The main and only difference between this model (with stochastic premium) and the Cramér-Lundberg or Classical Risk model is the way the premium comes in. Therefore, one would expect for a certain choice of parameters, this model would actually converge to the Cramér-Lundberg model. Suppose:

$$\left.\begin{matrix} \lambda^+ \to \infty \\ \beta \to \infty \end{matrix}\right\} \text{such that } \frac{\lambda^+}{\beta} \to c. \tag{3.10}$$

Now rewrite equation (3.9) in the following way:

$$\mathbb{E}[T_{x,f}] = \frac{\alpha+\beta}{\lambda^+ + \lambda^-} \cdot \frac{\lambda^+}{\lambda^+\alpha - \lambda^-\beta} \cdot \frac{\frac{1}{\beta}}{\frac{1}{\beta}} + \frac{(\alpha+\beta)^2}{(\lambda^+ + \lambda^-)^2} \cdot \frac{\lambda^+\lambda^-}{\lambda^+\alpha - \lambda^-\beta} \cdot \frac{\frac{1}{\beta}}{\frac{1}{\beta}} \cdot x$$

Now we know as a consequence of (3.10) that:

$$\lim_{\lambda^+,\,\beta\to\infty} \left(\frac{\alpha+\beta}{\lambda^+ + \lambda^-} \right) = \frac{1}{c}.$$

Therefore, we get the following result:

$$\mathbb{E}[T_{x,f}] = \frac{1}{c} \cdot \frac{c}{(c\alpha - \lambda^-)} + \frac{1}{c^2} \cdot \frac{\lambda^- cx}{(c\alpha - \lambda^-)} = \frac{c + \lambda^- x}{c(c\alpha - \lambda^-)}$$

Which is obviously the same as equation (2.17) as λ^- in this model equals λ in the Cramér-Lundberg model. So, as expected, this model converges to the Cramér-Lundberg model for a certain choice of parameters. We could even say that the Cramér-Lundberg is just a limiting case of the model with stochastic premium.

3.4 Simulation

Now that we have found an expression for expected time to ruin for this model with stochastic premium, we would like to see if a simulation will confirm the results found in the previous section. The main differences in the code, the results and similarities and differences with the Cramér-Lundberg simulation will be discussed in this section.

3.4.1 Code

The code that has been used to simulate this model is very similar to the code from the Cramér-Lundberg model. The same four classes have been used, making it unnecessary to elaborate on this again. However, due to the premium being stochastic rather than linear, twice as many random variable will need to be generated compared to the Cramér-Lundberg model. As Python is known to be relatively slow at generating single random variables, all random variables are now generated at once in the beginning of the algorithm. This will significantly speed up the code, especially when the premium arrival rates are chosen to be high, in order to see how this model converges to the Cramér-Lundberg model.

Algorithm 3 Model with Stochastic Premium Simulation

Initialize all parameters;
Initialize FES() and SimulateResults();
Initialize time and initial capital;
Generate $\lambda^+ \cdot T_{max}$ premium sizes and $\lambda^- \cdot T$ claim sizes;
Generate $\lambda^+ \cdot T_{max}$ inter-arrival times for the premiums;
Generate $\lambda^- \cdot T_{max}$ inter-arrival times for the claims;
Schedule arrival of first claim;
Schedule arrival of first premium;
while *Capital* ≥ 0 & *# Premiums* < $\lambda^+ \cdot T_{max}$ & *# Claims* < $\lambda^- \cdot T_{max}$ **do**
    Jump to the next event;
    Store the current time and capital;
    if *Next event is claim* **then**
        Subtract the claim amount from the capital;
        Store the time and capital;
        Schedule the next claim;
    end
    else if *Next event is premium* **then**
        Add Premium amount to the capital;
        Store the time and capital;
        Schedule next premium;
    end
end
return capital and time;

Hence, the main difference compared to the code of the Cramér-Lundberg model is the scheduling of premiums, which is now done in the same way as the claims are being scheduled. The algorithm we will use to generate results and confidence intervals for larger number of runs is the same as in the Cramér-Lundberg model.

3.4.2 Results

By plotting a single run again, we can verify that this model does what we expect. As there is no linear premium anymore, the plot will look different from Figure 2.1 and should only contain vertical and horizontal lines. We will take $\lambda^+ = \lambda^- = 1$, $x = 5$, $\mathbb{E}[Y_i] = 1.5$ and $\mathbb{E}[Z_j] = 2$ such that the safety loading condition is satisfied, as $1 \cdot 2 > 1 \cdot 1.5$, and the average income and expenses per time unit are the same as in Figure 2.1. We get the following plot:

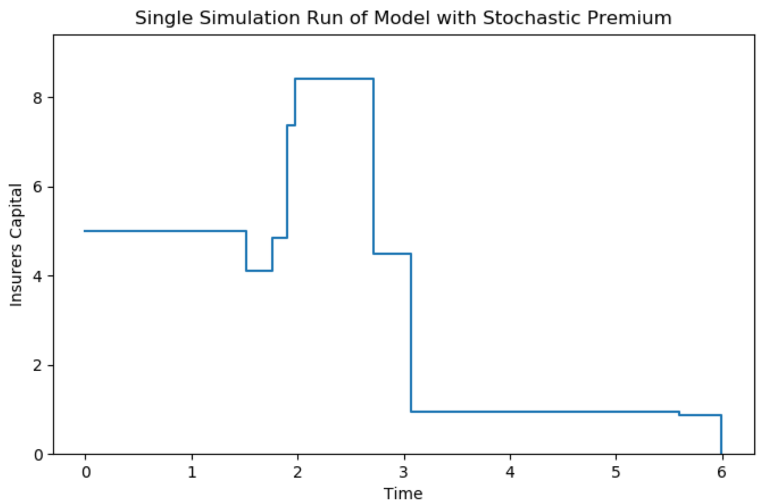


Figure 3.1: A single run of the model with stochastic premium.

As expected, the insurer's capital starts at $X(0) = 5$, then jumps up and down sometimes as a consequence of claims and premiums coming in and in this run, drops below 0 at $t \approx 6$. Now that we know the model works the way we want it to, we can start computing results a for larger number of runs.

As this model has more randomness compared to the Cramér-Lundberg, we will increase the number of runs to $N = 100000$ in order to get the following 95%-confidence intervals for the same parameters as in Figure 3.1:

Table 3.1: Confidence intervals of the model with stochastic premium for different initial capitals

Initial capital	$x = 0$	$x = 15$	$x = 30$
Time to Ruin	$(3.195, 3.711)$	$(32.999, 33.974)$	$(63.456, 65.991)$
Ruin Probability	$(0.862, 0.876)$	$(0.249, 0.255)$	$(0.0689, 0.0720)$

We can see that, even though the intervals for the ruin probability stay quite small, the intervals for the time to ruin grow bigger when the ruin probability gets smaller. Again, this is a consequence of the fact that we are only interested in the time to ruin for the cases where ruin actually occurs. When this happens only very few times, our time to ruin estimation will be less accurate. Nevertheless, we will not increase the number of runs due to time limitations.

We are interested in whether the results of the simulation match the theoretical findings. Furthermore, we want to visualize the convergence of the model with stochastic premium to the Cramér-Lundberg model. All this is done in the figure below. The safety loading condition θ is kept the same, while the premium arrival rate gets larger and the expected premium size smaller. The claim arrival rate λ^- and the expected claim size $\mathbb{E}[Y_i]$ are kept constant at $\lambda^- = 1$ and $\mathbb{E}[Y_i] = 1.5$.

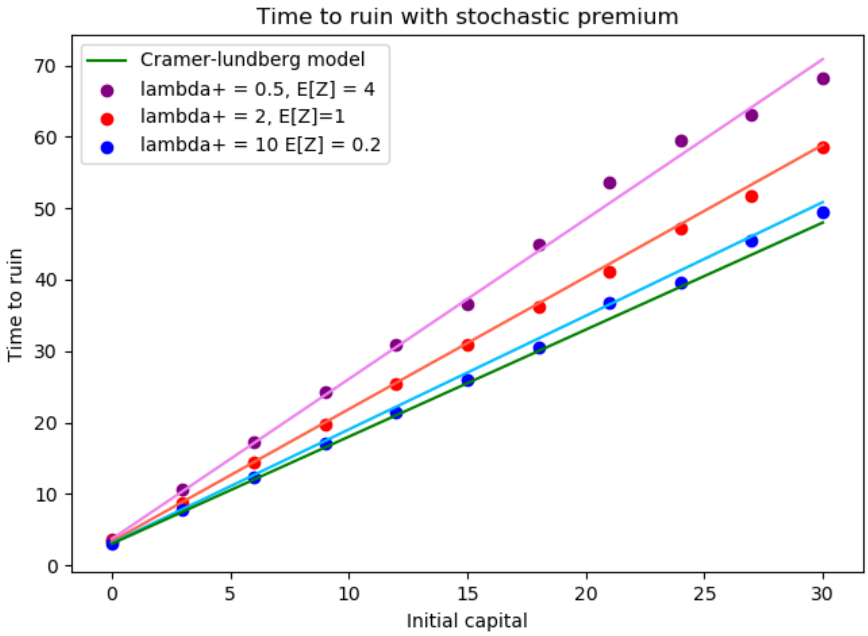


Figure 3.2: Time to ruin of the model with stochastic premium for different parameter sets.

The red, purple and blue lines represent the theoretical findings and the dots represent the simulation results. As can be seen in the legend of Figure 4.1, the red dots represent the simulation outcomes of the time to ruin for different initial capitals when the claim sizes are relatively high but arrive less frequent. This results into a steeper slope of the line, i.e. given that ruin does occur, it will generally take longer for it to occur when premiums arrive less frequent but are of a larger size, especially for higher initial capitals. We can explain this by taking into account that a certain number of claims need to arrive before the insurer's capital will drop below zero. When these claims arrive less frequently, it will therefore take longer for the capital to drop below zero, should that ever happen at all. When the expected premium size gets lower and the arrival rate increases, the model converges to the Cramér-Lundberg model. The theoretical results for the Cramér-Lundberg model are displayed by the green line, where the claim parameters are taken the same as for the model with stochastic premium and the premium rate is taken $c = 2$.

Therefore, not only via computations, but also through a stochastic simulation we can see that this model with stochastic premium converges to the Cramér-Lundberg model when the claim arrival rate grows large and the expected claim size gets small. Now that we have analyzed both models with linear premium and stochastic premium, we will combine these two to a model with both linear and stochastic premium in the next chapter.

4 Model with linear and stochastic premium

In this chapter we will combine the previous two chapters by introducing a model that can be seen as a combination of both the Cramér-Lundberg model and the model with stochastic premium. The claims still follow the same compound Poisson process, but the model has both linear and stochastic premium. Sabine Geurts has looked into the ruin probability for this model in her bachelor final project and we will again investigate the time to ruin. The way we will approach this is by first introducing an auxiliary model with hyper-exponentially distributed premium sizes and then choosing the parameters in such a way that this auxiliary model will converge to the model with linear and stochastic premium, similar to what has been done in Section 3.3.

4.1 Mathematical Model

This model contains elements of both the Cramér-Lundberg model as well as the model with stochastic premium. We again consider the capital of an insurance company at time t. At $t = 0$, we start with an initial surplus of x, which is assumed to be non-negative. The linear premium comes in at rate c, similar to the Cramér-Lundberg model. Hence, the total amount of (linear) premium received at time t equals $c \cdot t$. The stochastic premium comes in as a compound Poisson process: $P(t) := \sum_{j=1}^{N^+(t)} Z_j$, where $\{N^+(t)\}_{t\geq 0}$ is a Poisson process representing the number of premiums that occurred in the time interval $[0, t]$. Z_j is the amount of the j-th premium and is assumed to be non-negative for all $j \in \{1, 2, 3, ...\}$. The claims also follow a compound Poisson process, similar to the previous two models and again independent of the premiums. Hence, $S(t) := \sum_{i=1}^{N^-(t)} Y_i$, where $\{N^-(t)\}_{t\geq 0}$ with intensity λ^- represents the Poisson process that describes the number of claims that have arrived in the time interval $[0, t]$. As usual, Y_i represents the amount of the i-th claim and is assumed to be non-negative for all $i \in \{1, 2, 3, ...\}$. Therefore, the model looks as follows:

$$X(t) = x + ct + P(t) - S(t), \tag{4.1}$$

where $X(t)$ represents the capital of the insurance company at time t. We will again try to find an expression for the expected time to ruin, given that ruin occurs. As mentioned before, we introduce an auxiliary model that will help us computing an expression for the time to ruin for this model.

4.2 Auxiliary Model

As we have seen in Section 3.3, it is possible to choose the parameters of the compound Poisson premium process in such a way that it converges to a linear premium process. We did this to show that the Cramér-Lundberg model is actually a limiting case of the model with stochastic premium and in this chapter we will use this as a tool to find the expected time to ruin. Therefore, the auxiliary model has two types of stochastic premium, of which one will eventually converge to linear premium. As we assumed the claim sizes to be exponentially distributed in the previous chapters, we will do so again. Hence, the cumulative distribution function of the claims is given by $F(u) = \mathbb{P}(Y_i \leq u) = 1 - e^{-\alpha u}$. When we assume both the stochastic premium sizes to be exponentially distributed, the premium process of the auxiliary problem becomes hyper-exponential. We get the following model:

$$X_{\text{aux}}(t) = x + P_{\text{aux}}(t) - S(t), \tag{4.2}$$

where $P_{\text{aux}}(t)$ is defined as $P_{\text{aux}}(t) := \sum_{j=0}^{N^+(t)} W_j$, such that $G(v) = \mathbb{P}(W_j \leq v)$ is given by

$$G(v) := \begin{cases} 1 - e^{-\beta_1 v}, & \text{with probability } p_1, \\ 1 - e^{-\beta_2 v}, & \text{with probability } p_2. \end{cases}$$

In other words, $G(v) := p_1(1-e^{-\beta_1 v}) + p_2(1-e^{-\beta_2 v})$. By defining our auxiliary model this way, we can take the parameters such that $P_{\text{aux}}(t)$ converges to $ct + P(t)$. Note that this auxiliary model is actually the same model we have investigated in Chapter 3, but now with hyper-exponential premium sizes and exponential claim sizes.

The first thing we will do with this auxiliary model is determine the ruin probability. We will need both $\psi(x)$ and $\psi_1(x)$ to determine the expected time to ruin and as there is no expression for the time to ruin yet for this model, we will have to compute both. Therefore, let us start with finding an expression for the ruin probability.

Theorem 4.1. *The ruin probability, $\psi(x)$, of the auxiliary model, given by equation (4.2), is given by $\psi(x) = Ce^{-Rx}$, where R is the positive solution of the equation*

$$R^2 + \left(\frac{\lambda^+(p_1\beta_2 + p_2\beta_1 - \alpha) + \lambda^-(\beta_1+\beta_2)}{\lambda}\right) R - \left(\frac{\lambda^+\alpha(p_1\beta_2 + p_2\beta_1) - \lambda_1\beta_1\beta_2}{\lambda}\right) = 0,$$

and

$$C = \frac{\alpha - R}{\alpha}.$$

Proof. As the auxiliary model is a model with stochastic premium only, we assume that the ruin probability has the same form as it had in the model with stochastic premium, namely $\psi(x) = Ce^{-Rx}$, where $C, R \in \mathbb{R}$ are only dependent on the parameters of the claims and premiums. By conditioning on the first event we can find an integral equation of the time to ruin. We know that with probability $\frac{\lambda^+}{\lambda^+ + \lambda^-}$ the first event is a premium, which has probability $g(v)dv$ of being between size v and $v+dv$ for any $v \in (0,\infty)$, causing a jump in the surplus to level $x+v$. In this case, $g(v) = \mathbb{P}(G_i \le v) = p_1\beta_1 e^{-\beta_1 v} + p_2\beta_2 e^{-\beta_2 v}$. The probability that the first event is a claim is $\frac{\lambda^-}{\lambda^+ + \lambda^-}$ and with probability $f(u)du$ that claim is between size u and $u + du$. This claim can either be higher or lower than the initial capital. When u is higher than the initial capital, x, we have that the ruin probability $\psi(x-u) = 1$. When u is lower than the initial capital, this causes a jump in the surplus from x to $x - u$. All this combined gives the integral equation

$$\begin{aligned}\psi(x) = &\frac{\lambda^+}{\lambda}\int_0^\infty \psi(x+v)\left(p_1\beta_1 e^{-\beta_1 v} + p_2\beta_2 e^{-\beta_2 v}\right) dv \\ &+ \frac{\lambda^-}{\lambda}\int_0^x \psi(x-u)\alpha e^{-\alpha u} du + \frac{\lambda^-}{\lambda}\int_x^\infty 1\cdot\alpha e^{-\alpha u} du,\end{aligned} \tag{4.3}$$

where $\lambda = \lambda^+ + \lambda^-$. We now assume that $\psi(x) = Ce^{-Rx}$ and substitute this in equation (4.3). This gives the following equation:

$$\begin{aligned}Ce^{-Rx} = &\frac{\lambda^+}{\lambda}\int_0^\infty Ce^{-R(x+v)}\left(p_1\beta_1 e^{-\beta_1 v} + p_2\beta_2 e^{-\beta_2 v}\right) dv \\ &+ \frac{\lambda^-}{\lambda}\int_0^x Ce^{-R(x-u)}\alpha e^{-\alpha u} du + \frac{\lambda^-}{\lambda}\int_x^\infty \alpha e^{-\alpha u} du \\ = &\frac{\lambda^+}{\lambda}Ce^{-Rx}\left(\frac{p_1\beta_1}{R+\beta_1} + \frac{p_2\beta_2}{R+\beta_2}\right) + \frac{\lambda^-}{\lambda}Ce^{-Rx}\left(\frac{\alpha}{R-\alpha}(e^{(R-\alpha)x} - 1)\right) + \frac{\lambda^-}{\lambda}e^{-\alpha x}.\end{aligned}$$

Hence, when the ruin probability is given by $\psi(x) = Ce^{-Rx}$ we get that

$$Ce^{-Rx} = Ce^{-Rx}\left(\frac{\lambda^+}{\lambda}\left(\frac{p_1\beta_1}{R+\beta_1} + \frac{p_2\beta_2}{R+\beta_2}\right) - \frac{\lambda^-}{\lambda}\frac{\alpha}{R-\alpha}\right) + e^{-\alpha x}\left(C\frac{\lambda^-}{\lambda}\frac{\alpha}{R-\alpha} + \frac{\lambda^-}{\lambda}\right).$$

From this equation we can derive two equations, of which one will give us an expression for R and the other one of C. These equations are

$$\frac{\lambda^+}{\lambda}\left(\frac{p_1\beta_1}{R+\beta_1} + \frac{p_2\beta_2}{R+\beta_2}\right) - \frac{\lambda^-}{\lambda}\frac{\alpha}{R-\alpha} = 1, \tag{4.4}$$

$$C\frac{\lambda^-}{\lambda}\frac{\alpha}{R-\alpha} + \frac{\lambda^-}{\lambda} = 0. \tag{4.5}$$

We can verify that this method of substituting Ce^{-Rx} gives a valid outcome, as we can rewrite equation (4.4) in the following way:

$$\frac{\lambda^+}{\lambda}\left(\frac{p_1\beta_1}{\beta_1+R}+\frac{p_2\beta_2}{\beta_2+R}-1\right)+\frac{\lambda^-}{\lambda}\left(\frac{\alpha}{\alpha-R}-1\right)=0. \tag{4.6}$$

By noticing that the moment generating function of the exponentially distributed claim sizes is given by $m_{Y_i}(r)=\mathbb{E}[e^{rY_i}]=\frac{\alpha}{\alpha-r}$ and that the moment generating function of the hyper-exponentially distributed premium sizes evaluated at $-r$ is given by $m_{W_j}(-r)=\mathbb{E}[e^{-rW_j}]=\frac{p_1\beta_1}{\beta_1+r}+\frac{p_2\beta_2}{\beta_2+r}$, we can replace the expressions for the moment generating functions in equation (4.6) with the expectations and multiply the equation by λ, which will give us the equation

$$\lambda^+(\mathbb{E}[e^{-rW_j}]-1)+\lambda^-(\mathbb{E}[e^{rY_i}]-1)=0, \tag{4.7}$$

which happens to be exactly the adjustment equation of the model with stochastic premium in the bachelor final project of Sabine Geurts, found in Corollary 5.1 [3]. Hence, we know that when R is a positive root of adjustment equation (4.7), the upper-bound for the ruin probability $\psi(x)$ is equal to $\psi(x)\leq e^{-Rx}$. As we can derive from equation (4.5) that $C=\frac{\alpha-R}{\alpha}\leq 1$ when R is positive, we know that Ce^{-Rx} satisfies this upper-bound.

Now, we are left with finding an expression for R, using equation (4.4). We can rewrite this equation in the following way:

$$\frac{\lambda^+}{\lambda}(R-\alpha)\Big(p_1\beta_1(R+\beta_2)+p_2\beta_2(R+\beta_1)\Big)-\frac{\lambda^-}{\lambda}\alpha(R+\beta_1)(R+\beta_2)=(R+\beta_1)(R+\beta_2)(R-\alpha)$$

Using the fact that $p_1+p_2=1$, we can rewrite this equation as

$$R^3+\left(\frac{\lambda^+(p_1\beta_2+p_2\beta_1-\alpha)+\lambda^-(\beta_1+\beta_2)}{\lambda}\right)R^2-\left(\frac{\lambda^+\alpha(p_1\beta_2+p_2\beta_1)-\lambda^-\beta_1\beta_2}{\lambda}\right)R=0.$$

For obvious reasons, we do not have to take the solution $R=0$ into consideration. Therefore, we can divide this equation by R, which gives us the following quadratic equation:

$$R^2+\left(\frac{\lambda^+(p_1\beta_2+p_2\beta_1-\alpha)+\lambda^-(\beta_1+\beta_2)}{\lambda}\right)R-\left(\frac{\lambda^+\alpha(p_1\beta_2+p_2\beta_1)-\lambda^-\beta_1\beta_2}{\lambda}\right)=0. \tag{4.8}$$

The last thing we still need to do is to check whether this equation does have a positive solution. We know that when we take the limit $R\to\pm\infty$, the left hand side of the equation will go to infinity. However, when $R=0$, we get that the left hand side of the equation takes the value $-\frac{\lambda^+\alpha(p_1\beta_2+p_2\beta_1)-\lambda^-\beta_1\beta_2}{\lambda}$. If we can show that this value is smaller than 0, i.e. $\lambda^+\alpha(p_1\beta_2+p_2\beta_1)-\lambda^-\beta_1\beta_2>0$, we know that the equation has two solutions of which one is positive. Let us consider the safety loading condition of the auxiliary model:

$$\lambda^+\mathbb{E}[W_j]>\lambda^-\mathbb{E}[Y_i].$$

Substituting the expressions for the expected value of the premium and claim sizes gives us the following:

$$\lambda^+(\frac{p_1}{\beta_1}+\frac{p_2}{\beta_2})>\lambda^-\frac{1}{\alpha}$$

Which, when we multiply both sides with $\alpha\beta_1\beta_2$, can be rewritten as

$$\lambda^+\alpha(p_1\beta_2+p_2\beta_1)>\lambda^-\beta_1\beta_2, \text{ which implies that } \lambda^+\alpha(p_1\beta_2+_2\beta_1)-\lambda^-\beta_1\beta_2>0.$$

This is exactly what we were left with to show. So we have indeed that for the auxiliary model, given in equation (4.2), the ruin probability, $\psi(x)$, is given by $\psi(x)=Ce^{-Rx}$, where R is given by the positive solution of equation (4.8) and $C=\frac{\alpha-R}{\alpha}$. □

Now that we have found an expression for the ruin probability, $\psi(x)$, of this auxiliary model, we are left with finding an expression for $\psi_1(x) := \mathbb{E}[T_x \cdot \mathbb{1}(T_x < \infty)]$. Therefore, we introduce the following theorem:

Theorem 4.2. *Let $\psi(x)$ be given by Ce^{-Rx} (where C and R are given in Theorem 4.1), then we have that $\psi_1(x)$ is given by $\psi_1(x) = (A + Bx)e^{-Rx}$, where*

$$\begin{aligned} B &= \frac{C}{\lambda^- \frac{\alpha}{(\alpha-R)^2} - \lambda^+ \left(\frac{p_1\beta_1}{(R+\beta_1)^2} + \frac{p_2\beta_2}{(R+\beta_2)^2}\right)} \\ A &= \frac{B}{\alpha - R} \end{aligned} \tag{4.9}$$

Proof. As we have seen in the model with stochastic premium, the time to ruin is given by a linear function. We assume that this is not affected by a change from exponential to hyper-exponential premium sizes. Therefore, we start this proof by assuming that $\psi_1(x)$ is of the form $(A + Bx)e^{-Rx}$, so we have $\mathbb{E}[T_{x,f}] = \frac{\psi_1(x)}{\psi(x)} = \frac{A+Bx}{C}$. We will condition on the first event by using Theorem 3.1. Substituting $\psi_1(x) = (A + Bx)e^{-Rx}$ and $\psi(x) = Ce^{-Rx}$ gives us the equation

$$\begin{aligned} (A + Bx)e^{-Rx} = \frac{1}{\lambda}Ce^{-Rx} + \frac{\lambda^+}{\lambda}\int_0^\infty \big(A + B(x + v)\big)e^{-R(x+v)}\big(p_1\beta_1 e^{-\beta_1 v} + p_2\beta_2 e^{-\beta_2 v}\big)dv \\ + \frac{\lambda^-}{\lambda}\int_0^x \big(A + B(x - u)\big)e^{-R(x-u)}\,\alpha e^{-\alpha u}du. \end{aligned}$$

Before computing these integrals, we will split the first integral into two parts and rearrange some of the terms:

$$\begin{aligned} Ae^{-Rx} + Bxe^{-Rx} = \frac{1}{\lambda}Ce^{-Rx} &+ \frac{\lambda^+}{\lambda}e^{-Rx}p_1\beta_1\int_0^\infty \big(A + Bx + Bv\big)e^{-v(R+\beta_1)}dv \\ &+ \frac{\lambda^+}{\lambda}e^{-Rx}p_2\beta_2\int_0^\infty \big(A + Bx + Bv\big)e^{-v(R+\beta_2)}dv \\ &+ \frac{\lambda^-}{\lambda}e^{-Rx}\alpha\int_0^x \big(A + Bx - Bu\big)e^{-u(\alpha-R)}du \end{aligned} \tag{4.10}$$

Using partial integration, we get the following expressions for each of the integrals:

$$\begin{aligned} \int_0^\infty \big(A + Bx + Bv)\big)e^{-v(R+\beta_1)}dv &= \left[(A + Bx + Bv)\frac{-1}{R+\beta_1}e^{-v(R+\beta_1)}\right]_0^\infty + B\int_0^\infty \frac{1}{R+\beta_1}e^{-v(R+\beta_1)}dv \\ &= (A + Bx)\frac{1}{R+\beta_1} + B\frac{1}{(R+\beta_1)^2}. \end{aligned}$$

The same can be done with the second integral. Partial integration for the third integral will give the following:

$$\begin{aligned} \int_0^x \big(A + Bx - Bu)\big)e^{-u(\alpha-R)}du &= \left[(A + Bx - Bu)\frac{-1}{\alpha-R}e^{-u(\alpha-R)}\right]_0^x - B\int_0^x \frac{1}{\alpha-R}e^{-u(\alpha-R)}du \\ &= \left(\frac{A+Bx}{\alpha-R} - \frac{B}{(\alpha-R)^2}\right) + e^{-x(\alpha-R)}\left(\frac{B}{(\alpha-R)^2} - \frac{A}{\alpha-R}\right). \end{aligned}$$

We can substitute these expressions into the integral equation (4.10). After rearranging some of the terms, this gives us the following equation:

$$\begin{aligned} Ae^{-Rx} + Bxe^{-Rx} &= e^{-Rx}\Bigg(\frac{C}{\lambda} + \frac{\lambda^+}{\lambda}\Big(p_1\beta_1\big(\frac{A}{R+\beta_1} + \frac{B}{(R+\beta_1)^2}\big) + p_2\beta_2\big(\frac{A}{R+\beta_2} + \frac{B}{(R+\beta_2)^2}\big)\Big) \\ &+ \frac{\lambda^-}{\lambda}\alpha\Big(\frac{A}{\alpha-R} - \frac{B}{(\alpha-R)^2}\Big)\Bigg) + xe^{-Rx}\Bigg(\frac{\lambda^+}{\lambda}\Big(p_1\beta_1\frac{B}{R+\beta_1} + p_2\beta_2\frac{B}{R+\beta_2}\Big) + \frac{\lambda^-}{\lambda}\alpha\frac{B}{\alpha-R}\Bigg) \\ &+ e^{-\alpha x}\frac{\lambda^-}{\lambda}\alpha\Big(\frac{B}{(\alpha-R)^2} - \frac{A}{\alpha-R}\Big). \end{aligned}$$

From this equation, three expressions can be derived. These three expressions are given below:

$$A = \frac{C}{\lambda} + \frac{\lambda^+}{\lambda}\Big(p_1\beta_1\big(\frac{A}{R+\beta_1} + \frac{B}{(R+\beta_1)^2}\big) + p_2\beta_2\big(\frac{A}{R+\beta_2} + \frac{B}{(R+\beta_2)^2}\big)\Big) + \frac{\lambda^-}{\lambda}\alpha\Big(\frac{A}{\alpha-R} - \frac{B}{(\alpha-R)^2}\Big),$$
$$B = B\Big(\frac{\lambda^+}{\lambda}\Big(p_1\beta_1\frac{1}{R+\beta_1} + p_2\beta_2\frac{1}{R+\beta_2}\Big) + \frac{\lambda^-}{\lambda}\alpha\frac{1}{\alpha-R}\Big),$$
$$0 = \frac{\lambda^-}{\lambda}\alpha\Big(\frac{B}{(\alpha-R)^2} - \frac{A}{\alpha-R}\Big).$$

We will take these three equations into consideration to find expressions for A and B. First of all, we can see that the second equation is of no use, as this is in fact the same as equation (4.4). However, the first and the third equation will suffice for finding expressions for B and A, respectively. Let us consider the first equation. We can rewrite this in the following way:

$$A = \frac{C}{\lambda} + A\Big(\frac{\lambda^+}{\lambda}\Big(\frac{p_1\beta_1}{R+\beta_1} + \frac{p_2\beta_2}{R+\beta_2}\Big) + \frac{\lambda^-}{\lambda}\frac{\alpha}{\alpha-R}\Big) + B\Big(\frac{\lambda^+}{\lambda}\Big(\frac{p_1\beta_1}{(R+\beta_1)^2} + \frac{p_2\beta_2}{(R+\beta_2)^2}\Big) - \frac{\lambda^-}{\lambda}\frac{\alpha}{(\alpha-R)^2}\Big)$$

It can be observed that the part that is multiplied by A on the right hand side of this equation is again equal to equation (4.4). Using this result, we obtain the following expression for B:

$$B = \frac{C}{\lambda^- \frac{\alpha}{(\alpha-R)^2} - \lambda^+\left(\frac{p_1\beta_1}{(R+\beta_1)^2} + \frac{p_2\beta_2}{(R+\beta_2)^2}\right)}, \tag{4.11}$$

which concludes the first part of this proof. We are left with finding an expression for A. We can use the third equation to obtain that

$$\frac{A}{\alpha-R} = \frac{B}{(\alpha-R)^2},$$

which can be rewritten as

$$A = \frac{B}{\alpha-R}.$$

□

Now that we have found our expressions for $\psi(x)$ and $\psi_1(x)$, we can find an expression for the expected time to ruin. We know that $\mathbb{E}[T_{x,f}] = \frac{\psi_1(x)}{\psi(x)} = \frac{(A+Bx)e^{-Rx}}{Ce^{-Rx}} = \frac{A+Bx}{C}$. We define $G := \lambda^- \frac{\alpha}{(\alpha-R)^2} - \lambda^+\left(\frac{p_1\beta_1}{(R+\beta_1)^2} + \frac{p_2\beta_2}{(R+\beta_2)^2}\right)$, such that $B = \frac{C}{G}$. Then for the auxiliary model, we have that:

$$\mathbb{E}[T_{x,f}] = \frac{1}{G}\Big(x + \frac{1}{\alpha-R}\Big) \tag{4.12}$$

4.3 Convergence to Original Model

The next step is to see how these quantities converge when the parameters of the hyper-exponentially distributed claim sizes are chosen such that we get a model with linear and stochastic premium. Let us first take the parameter limits into consideration. For the hyper-exponentially distributed claim size to converge to both a linear premium and exponentially distributed claim size, the number of claims arriving per time unit will need to go to infinity, of which almost all claims will be of an infinitely small size. Only in some very rare cases, a claim of a larger, exponentially distributed size arrives. We will denote each parameter from the auxiliary model that will converge to some other parameter in the original model with a tilde and all parameters that appear in both the auxiliary and the original model without a tilde. We get the following parameter limits:

$$\left.\begin{aligned}\widetilde{\lambda^+} &\to \infty\\ \widetilde{p_2} &\to 0\end{aligned}\right\} \text{such that } \widetilde{\lambda^+}\cdot\widetilde{p_2} \to \lambda^+,$$
$$\widetilde{\beta_2} \to \beta,$$
$$\left.\begin{aligned}\widetilde{p_1} &\to 1\\ \widetilde{\beta_1} &\to \infty\end{aligned}\right\} \text{such that } \frac{\widetilde{\lambda^+}\cdot\widetilde{p_1}}{\widetilde{\beta_1}} \to c.$$

The first thing we will consider is the quadratic equation of R, equation (4.8). Using the parameter limits, we can get a quadratic equation for the model with linear and stochastic premium:

$$R^2 + \frac{c\widetilde{\beta_1}\beta + \lambda^+\widetilde{\beta_1} - \widetilde{\lambda^+}\alpha + \lambda^-\widetilde{\beta_1} + \lambda^-\beta}{\widetilde{\lambda^+}+\lambda^-}R - \frac{\alpha c\widetilde{\beta_1}\beta + \lambda^+\alpha\widetilde{\beta_1} - \lambda^-\widetilde{\beta_1}\beta}{\widetilde{\lambda^+}+\lambda^-} = 0$$

$$R^2 + \frac{\widetilde{\beta_1}\big(c\beta + \lambda^+ + \lambda^-\big) - \alpha\widetilde{\lambda^+} + \lambda^-\beta}{\widetilde{\lambda^+}+\lambda^-}R - \frac{\widetilde{\beta_1}\big(\alpha c\beta + \lambda^+\alpha - \lambda^-\beta\big)}{\widetilde{\lambda^+}+\lambda^-} = 0$$

$$R^2 + \big(-\alpha + \frac{1}{c}(c\beta+\lambda)\big)R - \frac{1}{c}\big(\alpha\beta c + \lambda^+\alpha - \lambda^-\beta\big) = 0$$

$$R^2 - \big(\alpha - \beta - \frac{\lambda}{c}\big)R - \alpha\beta - \frac{1}{c}\big(\lambda^+\alpha - \lambda^-\beta\big) = 0.$$

This equation matches the quadratic equation Sabine Geurts has found in her bachelor final project [3] for the exponent of the time to ruin. We then also know that the positive solution of the quadratic equation of R is given by

$$R = \frac{1}{2}\Big(\alpha - \beta - \frac{\lambda}{c} + \sqrt{\big(\alpha-\beta-\frac{\lambda}{c}\big)^2 + 4\big(\alpha\beta + \frac{1}{c}(\lambda^+\alpha - \lambda^-\beta)\big)}\,\Big). \tag{4.13}$$

Now let us take the term G into consideration. We know that G has been defined as

$$\widetilde{G} = \lambda^-\frac{\alpha}{(\alpha-R)^2} - \widetilde{\lambda^+}\left(\frac{\widetilde{p_1}\widetilde{\beta_1}}{(R+\widetilde{\beta_1})^2} + \frac{\widetilde{p_2}\widetilde{\beta_2}}{(R+\widetilde{\beta_2})^2}\right)$$

Again, taking the parameter limits will give us an expression for G in terms of the model with linear and stochastic premium. We get the equation:

$$\frac{\lambda^-\alpha}{(\alpha-R)^2} - \frac{c(\widetilde{\beta_1})^2}{(R+\widetilde{\beta_1})^2} - \frac{\lambda^+\beta}{(R+\beta)^2}$$

and hence we have:

$$G = \frac{\lambda^-\alpha}{(\alpha-R)^2} - c - \frac{\lambda^+\beta}{(R+\beta)^2}. \tag{4.14}$$

All in all, we now have found an expression for the time to ruin of the model with linear and stochastic premium, given by the following formula:

$$\mathbb{E}[T_{x,f}] = \frac{1}{G}\Big(x + \frac{1}{\alpha - R}\Big), \tag{4.15}$$

where G is given by equation (4.14) and R is given by (4.13).

Now that we have found an analytic expression for the time to ruin, we will validate it by writing a stochastic simulation for this model with both linear and stochastic premium and use this to compute the expected time to ruin.

4.4 Simulation

We will use simulation to verify whether these results are correct. It might be interesting to see what the effect on the time to ruin is when there is relatively more stochastic or linear premium. We would expect this model to start behaving like the Cramér-Lundberg model when most of the premium comes in linearly. When there is relatively more stochastic premium, the time to ruin of this model will mostlikely give similar values to the model with solely stochastic premium.

4.4.1 Code

Similar to the model with stochastic premium only, this model has twice as many random events happening as the Cramér-Lundberg model. Therefore, we will generate all random variables at the start of our code, making the simulation significantly more efficient. The only difference from this code compared to the code of the model with stochastic premium is that we now keep track of the linear premium, updating this every time an event happens. The pseudo-code is given below, where the differences with the code of the model with stochastic premium are highlighted.

Algorithm 4 Model with Stochastic and Linear Premium Simulation

```
Initialize all parameters;
Initialize FES() and SimulateResults();
Initialize time and initial capital;
Generate λ+ · T_max premium sizes and λ− · T claim sizes;
Generate λ+ · T_max inter-arrival times for the premiums;
Generate λ− · T_max inter-arrival times for the claims;
Schedule arrival of first claim;
Schedule arrival of first premium;
while Capital ≥ 0 & # Premiums < λ+ · T_max & # Claims < λ− · T_max do
    Jump to the next event;
    Update the linear capital increase;
    Store the current time and capital;
    if Next event is claim then
        Subtract the claim amount from the capital;
        Store the time and capital;
        Schedule the next claim;
    end
    else if Next event is premium then
        Add Premium amount to the capital;
        Store the time and capital;
        Schedule next premium;
    end
end
return capital and time;
```

As can be seen in the pseudo-code, there is only one line different from the code of the model with stochastic premium. However, as we have seen in the previous part of this chapter, this linear premium does have a significant effect on the expression for the time to ruin. We will use this simulation to see how both the linear and stochastic premium affect the results.

4.4.2 Results

When plotting a single run of this model with both linear and stochastic premium, we expect the plot to look like the model with stochastic premium, only now the horizontal lines will be increasing at rate c. In the figure below, a single run where ruin occurs has been plotted. For this run, we have again set the initial capital to $x = 5$. The parameters for the premiums have been set to $c = 1$, $\lambda^+ = 1$ and $\mathbb{E}[Z_i] = 1$, such that the total income per time unit is equal to $c + \lambda^+ \cdot \mathbb{E}[Z_j] = 1 + 1 \cdot 1 = 2$. The parameters for the claims have been set to $\lambda^- = 1$ and $\mathbb{E}[Y_i] = 1.5$, such that total expenses per time unit are equal to $\lambda^- \cdot \mathbb{E}[Y_i] = 1 \cdot 1.5 = 1.5$, satisfying the safety loading condition. Hence, the income and expenses per time unit are the same as in the single runs of both the previous models.

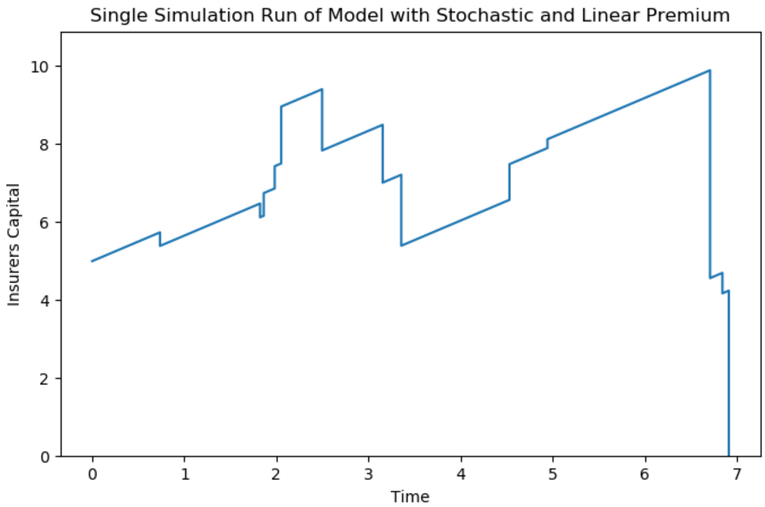


Figure 4.1: Single run of the model with both linear and stochastic premium.

It can be seen in this plot that the insurer's capital acts the way we would expect it to. That is, with an upward slope and occasional jumps up and down. Therefore, we can use this code to compute the expected time to ruin numerically using a large number of runs. For the same initial capitals as in previous chapters, the 95%-confidence intervals of the expected time to ruin and ruin probabilities with $N = 100000$ runs are given in the table below.

Table 4.1: Confidence intervals of the model with stochastic and linear premium for different initial capitals

Initial capital	$x = 0$	$x = 15$	$x = 30$
Time to Ruin	$(3.1291, 3.2558)$	$(28.545, 29.652)$	$(52.868, 57.364)$
Ruin Probability	$(0.7937, 0.7987)$	$(0.1041, 0.1079)$	$(0.0126, 0.0141)$

Similar to the confidence table of the model with stochastic premium, the confidence intervals for the time to ruin get larger when the initial capital increases, due to a decrease in the number of runs where ruin occurs when the initial capital gets larger. When we compare these values to the values of Table 4.1, the time to ruin is significantly smaller for this model, especially when the initial capital gets higher. We will also see this in the plots later in this section.

As this model is a combination of the Cramér-Lundberg model and the model with stochastic premium, we are interested in which part of the expected time to ruin is a result of the linear premium and what part is a result of the stochastic premium. In order to visualize this, we will keep the safety loading fixed, at $\theta = \frac{1}{3}$, as well as the parameters for the claims, $\lambda^- = 1$ and $\mathbb{E}[Y_i] = 1.5$ and the rate for the stochastic premium, $\lambda^+ = 1$. Varying the premium from being almost completely stochastic to almost completely linear shows us how this shift in type of premium influences the expected time to ruin. We get the following plot:

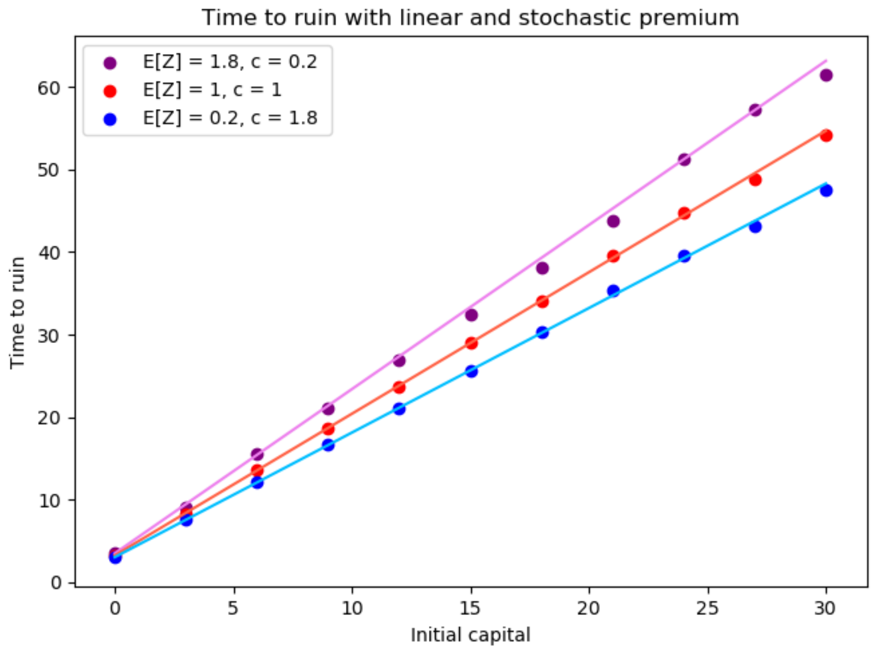


Figure 4.2: Differences in the time to ruin when the premium is shifted from linear to stochastic.

In a way, this plot visualizes the transition from the Cramér-Lundberg model to the model with stochastic premium. It can be seen that when most of the premium is linear, the expected time to ruin grows less quickly compared to when the premium is mostly stochastic. This is a result of the fact that when the premium is mostly stochastic, the insurer's capital evidently grows relatively slow. Due to the fact that the insurance company has to wait longer before it receives a noteworthy amount of premium, instead of receiving it at a constant rate, their capital will not grow as fast and hence ruin will occur more often and with a higher probability at a later stage. Hence, we can conclude that, from the insurance's point of view, having a constant inflow of premium is more beneficial than receiving it at certain unknown points in time.

Now we have seen the three types of insurance models that were all treated in the bachelor final project of Sabine Geurts [3]. For each of these models, an expression for the expected time to ruin has been found and a simulation was written. The plots in the simulation section of each chapter show that for each of the models, the analytic expressions match the numerical results, implying the results are all correct and no mistakes have been made. All three of these models have variations in the way the premium is modelled, but have the same way of modeling the claims. Therefore, we will look at one more model which has an underlying Markov-model for the claims.

5 Model with Markovian States

So far, we have computed the expected time to ruin for a model with linear premium (Cramér-Lundberg model), a model with stochastic premium and a model with both linear and stochastic premium. However, the claims have been modelled the exact same way for each model, namely as a compound Poisson process with exponentially distributed claim sizes. Therefore, the main focus of this chapter is to introduce a new model that will model the claims in a different way than we have seen in the previous three chapters.

5.1 Mathematical Model

Up and until now, the expected claim size and the rate at which the claims come in have always been fixed constants. However, in reality these quantities might change over time. A travel insurance company, for example, will have to pay much more claims at peak season compared to off-season. Similar fluctuations in claim sizes and the claim arrival rate might occur with other types of insurance companies. Hence, the model that will be discussed in this chapter will have stochastic premiums and claims, but the parameters of the claim process will change over time according to the states of an underlying Markov process. In other words, at some moments in time there might be relatively few claims, while at other moments a lot more claims come in. On the other hand, some periods the average claim size is relatively low, while it is significantly higher at other periods in time. In this model, we will assume the state space S of the underlying Markov model consists of two states only, say state G and state B, to which, from now on, we will also refer to as the good and the bad state, respectively. As in this model, the underlying Markov process is given by a continuous time Markov chain, we know that the process will change states from one state to another after holding time H_i; an exponentially distributed random variable where $i \in \{G, B\}$ represents the current state. Both H_G and H_B are independent, such that $H_G \sim Exp(\gamma)$ and $H_B \sim Exp(\nu)$. After the holding time, a transition is made from one state to the other one with probability 1, as there are only two different states. Besides this underlying Markov process for the claims, the model will be the exact same model as we have seen in Chapter 3. Hence, the model looks as follows:

$$X(t) = x + P(t) - S(t), \tag{5.1}$$

where $X(t)$ is the insurer's capital at time t, x is the insurer's capital at $t = 0$, $P(t)$ is the total premium amount at time t, given by $\sum_{j=1}^{N^+(t)} Z_j$, where $\{N^+(t)\}_{t\geq 0}$ is the Poisson process with intensity λ^+, describing the number of premiums that have arrived within the time interval $[0, t]$ and Z_j is the size of the j-th premium, with $j \in \{1, 2, 3, ...\}$ and with cumulative distribution function $G(v) = \mathbb{P}(Z_j \leq v) = 1 - e^{-\beta v}$. The total claim amount at time t is given by $S(t) := \sum_{i=1}^{N^-(t)} Y_i$ where Y_i is the size of the i-th claim, with $i \in \{1, 2, 3, ...\}$ and with cumulative distribution function $F(u) = \mathbb{P}(Y_i \leq u) = 1 - e^{-\alpha_B u}$ when the Markov process is in state B and $F(u) = 1 - e^{-\alpha_G u}$ when the Markov process is in state G. Now $\{N^-(t)\}_{t\geq 0}$ is the Markov-modulated Poisson process that describes the number of claims within the time interval $[0, t]$ and has rate λ_B^- when the Markov process is in state B and has rate λ_G^- when the Markov process is in state G.

The aim of this chapter is to find out whether this variation in modeling the claims affects the general form of the expressions for the ruin probability and the expected time to ruin. In other words, we want to know if the expressions for the ruin probability and the expected time to ruin are still exponentially decreasing and linearly increasing, respectively. If so, we will try to find these expressions and see how they differ compared to the previous models.

5.2 Ruin Probability

We will start the computations on the ruin probability again by conditioning on the first claim. However, we have to distinguish two cases, namely the case where the Markov process starts in state G and the case where it starts in state B. Suppose it starts in state G. Then there are 3 events that can happen after an exponentially distributed amount of time: a premium comes in, a claim comes in or there is a switch in the Markov process from state G to state B. The probabilities for these events to take place are $\frac{\lambda^+}{\lambda^++\lambda_G^-+\gamma}$, $\frac{\lambda_G^-}{\lambda^++\lambda_G^-+\gamma}$ and $\frac{\gamma}{\lambda^++\lambda_G^-+\gamma}$, respectively. When a premium comes in, it has probability $g(v)dv$ of being of a size between v and $v+dv$ and can be of any size between 0 and ∞, causing a jump in the surplus from x to $x+v$. When a claim comes in, it has probability $f(u)du$ of being of a size between u and $u+du$ and can either be higher or lower than the initial capital, x. When it is higher then the initial capital, it will result into ruin. When the claim is lower than the initial capital, it will cause a jump in the surplus from x to $x-u$. When the Markov process switches from state G to state B, this whole process starts again but now considered from state B. Therefore, we get the following integral equation for the ruin probability when beginning in state G:

$$\begin{aligned}\psi_G(x) = {}& \frac{\gamma}{\lambda^+ + \lambda_G^- + \gamma}\psi_B(x) + \frac{\lambda^+}{\lambda^+ + \lambda_G^- + \gamma}\int_0^\infty \psi_G(x+v)\beta e^{-\beta v}dv \\ &+ \frac{\lambda_G^-}{\lambda^+ + \lambda_G^- + \gamma}\int_0^x \psi_G(x-u)\alpha_G e^{-\alpha_G u}du + \frac{\lambda_G^-}{\lambda^+ + \lambda_G^- + \gamma}\int_x^\infty \alpha_G e^{-\alpha_G u}du.\end{aligned} \tag{5.2}$$

Taking the same steps when the initial state is B, gives the integral equation

$$\begin{aligned}\psi_B(x) = {}& \frac{\nu}{\lambda^+ + \lambda_B^- + \nu}\psi_G(x) + \frac{\lambda^+}{\lambda^+ + \lambda_B^- + \nu}\int_0^\infty \psi_B(x+v)\beta e^{-\beta v}dv \\ &+ \frac{\lambda_B^-}{\lambda^+ + \lambda_B^- + \nu}\int_0^x \psi_G(x-u)\alpha_B e^{-\alpha_B u}du + \frac{\lambda_B^-}{\lambda^+ + \lambda_B^- + \nu}\int_x^\infty \alpha_B e^{-\alpha_B u}du.\end{aligned} \tag{5.3}$$

Again, we will assume that the ruin probability of both models is given by an exponential function of the form Ce^{-Rx}. We do not know yet whether the exponent of both expressions will be the same, hence, for now we assume that $\psi_G(x) = C_G e^{-R_G x}$ and $\psi_B(x) = C_B e^{-R_B x}$, where $C_G, C_B, R_G, R_B \in \mathbb{R}^+$. When we substitute this in equations (5.2) and (5.3), we can write the integral equation of the ruin probability when beginning in state G in in the following way:

$$\begin{aligned}C_G e^{-R_G x} = {}& \frac{\gamma}{\lambda^+ + \lambda_G^- + \gamma}C_B e^{-R_B x} + \frac{\lambda^+}{\lambda^+ + \lambda_G^- + \gamma}\int_0^\infty C_G e^{-R_G(x+v)}\beta e^{-\beta v}dv \\ &+ \frac{\lambda_G^-}{\lambda^+ + \lambda_G^- + \gamma}\int_0^x C_G e^{-R_G(x-u)}\alpha_G e^{-\alpha_G u}du + \frac{\lambda_G^-}{\lambda^+ + \lambda_G^- + \gamma}\int_x^\infty \alpha_G e^{-\alpha_G u}du.\end{aligned}$$

The ruin probability when beginning in state B looks as follows:

$$\begin{aligned}C_B e^{-R_B x} = {}& \frac{\nu}{\lambda^+ + \lambda_B^- + \nu}C_G e^{-R_G x} + \frac{\lambda^+}{\lambda^+ + \lambda_B^- + \nu}\int_0^\infty C_B e^{-R_B(x+v)}\beta e^{-\beta v}dv \\ &+ \frac{\lambda_B^-}{\lambda^+ + \lambda_B^- + \nu}\int_0^x C_B e^{-R_B(x-u)}\alpha_B e^{-\alpha_B u}du + \frac{\lambda_B^-}{\lambda^+ + \lambda_B^- + \nu}\int_x^\infty \alpha_B e^{-\alpha_B u}du.\end{aligned}$$

Now let us define $g := \lambda^+ + \lambda_G^- + \gamma$ and $b := \lambda^+ + \lambda_B^- + \nu$. We can compute the integrals from both equations to obtain the following expressions on the ruin probability:

$$\begin{aligned}C_G e^{-R_G x} - \frac{\gamma}{g}C_B e^{-R_B x} = {}& C_G e^{-R_G x}\left(\frac{\lambda^+}{g}\frac{\beta}{R_G+\beta} + \frac{\lambda_G^-}{g}\frac{\alpha_G}{\alpha_G - R_G}\right) \\ &+ \frac{\lambda_G^-}{g}e^{-\alpha_G x}\left(1 - C_G\frac{\alpha_G}{\alpha_G - R_G}\right),\end{aligned}$$

$$C_B e^{-R_B x} - \frac{\nu}{b} C_G e^{-R_G x} = C_B e^{-R_B x} \left(\frac{\lambda^+}{b} \frac{\beta}{R_B + \beta} + \frac{\lambda_B^-}{b} \frac{\alpha_B}{\alpha_B - R_B} \right) + \frac{\lambda_B^-}{b} e^{-\alpha_B x} \left(1 - C_B \frac{\alpha_B}{\alpha_B - R_B} \right).$$

From these equations we can conclude that $R_G = R_B$, as otherwise we would have that $\frac{\gamma}{g} C_B = \frac{\nu}{b} C_G = 0$, which is not possible as we know that $\frac{\gamma}{g}$, $\frac{\nu}{b}$, C_G and C_B are all greater than zero. Therefore, we have that $\psi_G(x) = C_G e^{-Rx}$ and $\psi_B(x) = C_B e^{-Rx}$, for some $R \in \mathbb{R}^+$. Using this, we can derive the following two equations

$$(C_G - \frac{\gamma}{g} C_B) e^{-Rx} = C_G e^{-Rx} \left(\frac{\lambda^+}{g} \frac{\beta}{R + \beta} + \frac{\lambda_G^-}{g} \frac{\alpha_G}{\alpha_G - R} \right) + \frac{\lambda_G^-}{g} e^{-\alpha_G x} \left(1 - C_G \frac{\alpha_G}{\alpha_G - R} \right),$$

$$(C_B - \frac{\nu}{b} C_G) e^{-Rx} = C_B e^{-Rx} \left(\frac{\lambda^+}{b} \frac{\beta}{R + \beta} + \frac{\lambda_B^-}{b} \frac{\alpha_B}{\alpha_B - R} \right) + \frac{\lambda_B^-}{b} e^{-\alpha_B x} \left(1 - C_B \frac{\alpha_B}{\alpha_B - R} \right).$$

Let us consider the first equation with begin state G. We can can derive two equations from this expression, namely:

$$C_G - \frac{\gamma}{g} C_B = C_G \left(\frac{\lambda^+}{g} \frac{\beta}{R + \beta} + \frac{\lambda_G^-}{g} \frac{\alpha_G}{\alpha_G - R} \right) \tag{5.4}$$

and

$$1 - C_G \frac{\alpha_G}{\alpha_G - R} = 0. \tag{5.5}$$

From equation (5.5) we can derive that $C_G = \frac{\alpha_G - R}{\alpha_G}$. The same can be done with the equation with begin state B, to derive an expression for C_B, namely: $C_B = \frac{\alpha_B - R}{\alpha_B}$. However, when we would substitute these two expressions in equation (5.4), we will find an expression for R which does not depend on the holding time and the rate of the claim arrival process of state B, given by ν and λ_B^-, respectively. On the other hand, when we take the same steps but for the case with begin state B, we will get a different expression which does depend on λ_B^- and ν, but does not depend on λ_G^- and γ. Assuming no mistakes have been made in the above computations, we can conclude that the ruin probability for an insurance model with an underlying Markov process is generally not given by an expression of the form Ce^{-Rx}.

There is some literature on this topic, in particular a book by S. Asmussen and H. Albrecher named "Ruin Probabilities" [5], where risk theory in a Markovian environment is investigated. It is shown that, when starting in state i, the ruin probability of the Cramér-Lundberg model in a Markovian environment is not simply of the form $C_i e^{-Rx}$, but that there is an expectation of other quantities added to this formula. As we have seen that the Cramér-Lundberg model is a limiting case of the model with stochastic premium, we know that also for the model discussed in this chapter, the ruin probability is not of the form $C_i e^{-Rx}$, which explains why we were not able to find an expression for the ruin probability by substituting this formula. As the theory behind such models with Markov-modulated claims turns out to be quite complex, we will no longer try to find an expression for the expected time to ruin analytically, but observe it using a stochastic simulation.

5.3 Simulation

Even though we were not able to find an expression for the ruin probability and the time to ruin for this model, we can still perform a simulation to see if this gives us any interesting insights. We can, for example, observe how the begin state affects the ruin probability and the time to ruin and also check whether the ruin probability does indeed look like an exponentially decreasing function. First of all, we will need a new code that simulates the underlying Markov process determining the claim parameters.

5.3.1 Code

The code will have a copious amount of overlap with the code of the model with stochastic premium. However, we will have to implement an underlying Markov-process that affects the size and the frequency of the arriving claims. In the pseudo-code below, the algorithm that has been used is explained in detail:

Algorithm 5 Model with Markovian States

```
Initialize all parameters;
Initialize FES() and SimulateResults();
Initialize time and initial capital;
Determine initial state, either random or not;
if Initial state is 'good' then
    Schedule first claim according to good-state-parameters;
    Schedule first state switch to 'bad' state;
end
else
    Schedule first claim according to bad-state-parameters;
    Schedule first state switch to 'good' state;
end
Schedule first premium;
while Capital ≥ 0 & time < T_max do
    Jump to the next event;
    Store the current time and capital;
    if Next event is claim then
        Subtract claim amount from capital;
        Store the time and capital;
        if State is 'good' then
            Schedule next claim according to good-state-parameters;
        end
        else
            Schedule next claim according to bad-state-parameters;
        end
    end
    else if Next event is premium then
        Add Premium amount to the capital;
        Store the time and capital;
        Schedule next premium;
    end
    else if Next event is state switch then
        Switch state from bad to good or vice versa;
        if state is good then
            Schedule new state switch from good to bad;
        end
        else
            Schedule new state switch from bad to good;
        end
    end
end
return capital and time;
```

First of all we will have to determine in which state we begin. We can do this manually, which is useful when we are interested in how the begin state affects the expected time to ruin, or we can do this at random. Furthermore, we will have one new type of event besides a claim or premium arriving, namely a "state-switch". When this event happens, claims will be generated with a different arrival rate and expected claim size, according to the state the Markov process is in at that moment. Note that for now, we do not adjust the arrival time and the size of claims that have already been scheduled, when there is a transition of one state to the other right before this claim. This would very much increase the simulation time as we would have to generate new random variables every time a state-switch event arrives. Hence, when both the claim arrival rate and the holding time are chosen sufficiently large, we assume the effect this way of programming has on the outcome is negligibly small.

As can be seen, the code has become a bit longer. However, this is mainly caused by the fact that we have to repeatedly check in what state the Markov process is and schedule the new events accordingly. We will use this code to generate results for both the ruin probability and the expected time to ruin. The algorithm that is used for generating the confidence intervals and running the algorithm for a large number of runs is the same as in the previous models.

5.3.2 Results

The first thing we will do is what we have done similarly in the previous chapters, namely running one single run. We would expect to see certain trends in the insurer's capital up and down according to the state the Markov process is in, especially when the safety loading condition is satisfied in the good state, but not in the bad state. Therefore, we will use the following parameters: $\gamma = \nu = 0.05$, hence the holding time of both states is equal to 20 time units, $\mathbb{E}[Z_j] = 2$, $\lambda^+ = 1$, so the total income per time unit is $\lambda^+ \cdot \mathbb{E}[Z_j] = 2$. For the claims we have the following parameters for the bad state (B) and the good state (G), respectively: $\mathbb{E}[Y_i]_B = 1.5$, $\lambda_B^- = 2$ and $\mathbb{E}[Y_i]_G = 1$, $\lambda_G^- = 1$. Hence, the safety loading condition is satisfied when the Markov process is in the good state, but it is not when the Markov process is in the bad state. When we start in the bad state with an initial capital of $x = 10$ and set $T_{max} = 200$, we get the following figure:

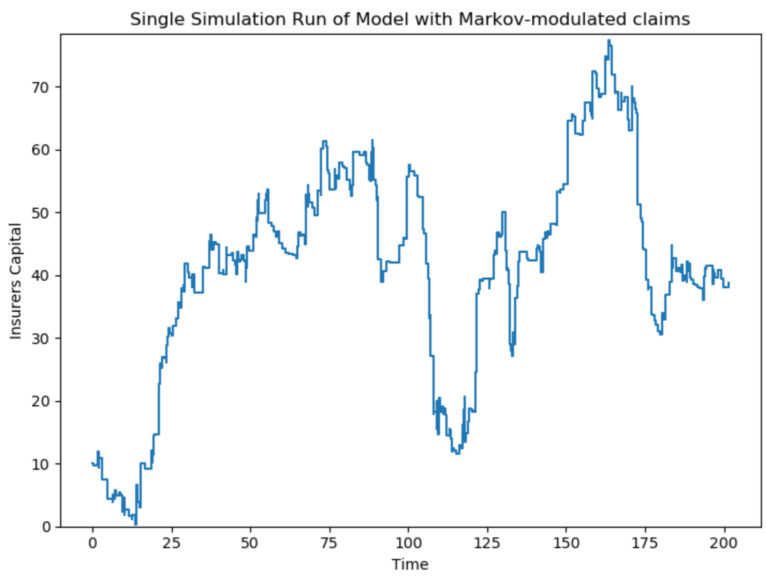
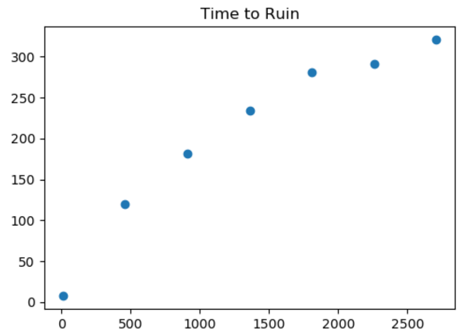


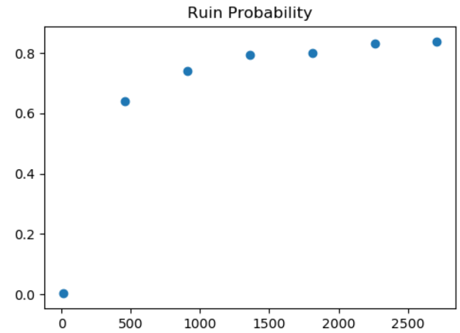
Figure 5.1: Single run of the model with stochastic premium and Markovian states for the claims.

In this plot we can easily distinct the trends of the insurer's capital. It looks like the Markov process switched to the good state at around $t = 15$ and, according to the choice of the Markov process parameters, μ and γ, switched in total about 10 times within the time frame of this plot.

Now that we have seen that the model behaves the way we would expect it, we can start generating results for a larger number of runs. As can be seen in Figure 5.1, the insurer's capital does not grow as fast for this model as it did for the previous models. Hence, when choosing a value for T_{max}, it might be useful to make a plot like Figure 2.2, for both the ruin probability as the time to ruin, to see what should be the value for T_{max} when the parameters are chosen as they are in Figure 5.1 and the initial capital is equal to $x = 30$. These plots look as follows:



(a) Expected time to ruin



(b) Ruin probability

Figure 5.2: Influence of T_{max} on the time to ruin and the ruin probability for $x = 30$.

In the two figures above there are a few things that stand out. First of all, the ruin probability seems to have reached a relatively stable value when T_{max} is around 1500, a value that is significantly higher compared to the previous models. The expected time to ruin, however, is still increasing at $T_{max} = 2500$. Furthermore, the value for the time to ruin as well as the value for the ruin probability is much higher than it was for the previous models at $x = 30$, regardless of the choice of parameters of those models. All the above is a consequence of the fact that the insurer's capital does not grow fast enough to make sure it reaches a point where ruin does not occur anymore. Even after a lot of time, the insurance company still has a relatively high probability of going bankrupt, while at the other models this did barely occur after a certain value for T_{max}. Therefore, for the remainder of this section, we will choose the parameters of both the good and the bad state such that the safety loading condition is satisfied at all times. This allows us to pick a feasible value for T_{max} and compute accurate results for a large number of runs.

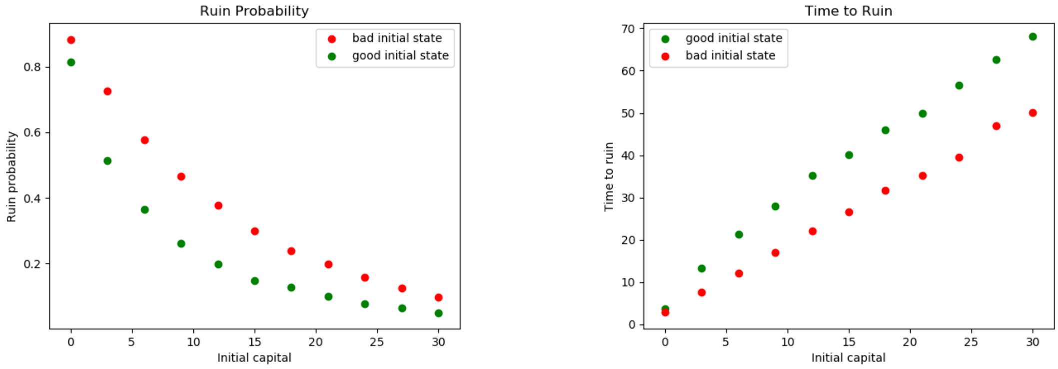
As we have done before, we will start with generating some 95%-confidence intervals to see how accurate the numerical results are for different initial capitals and both a good and a bad begin state. We will take all parameters the same as they were in Figure 5.1, except for the claim parameters for the bad state, which we will take equal to $\lambda_B^+ = 1$ and $\mathbb{E}[Y_i]_B = 1.8$, such that the safety loading condition is also satisfied when the Markov process is in the bad state. These parameters give the following confidence intervals for $N = 100000$ runs:

Table 5.1: Confidence intervals for the model with an underlying Markov process for different initial capitals.

Initial capital	$x = 0$	$x = 15$	$x = 30$
Time to ruin (good begin state)	$(3.263, 3.435)$	$(40.052, 41.415)$	$(64.243, 67.200)$
Ruin probability (good begin state)	$(0.8338, 0.8384)$	$(0.1557, 0.1602)$	$(0.0495, 0.0522)$
Time to ruin (bad begin state)	$(3.326, 3.478)$	$(27.592, 28.391)$	$(50.917, 52.878)$
Ruin probability (bad)	$(0.8727, 0.8768)$	$(0.2822, 0.2878)$	$(0.0929, 0.0965)$

We can see that these values for the expected time to ruin and the ruin probability are way more consistent with the results of the previous models than the values in Figure 5.2. Obviously, the higher the ruin probability, the more accurate the results are.

The first thing we are interested in is what the effect of the initial state is on the ruin probability and the time to ruin. For the same parameters as in the table and different initial capitals and initial states, the time to ruin and ruin probability are plotted in Figure 5.3.



(a) Ruin probability

(b) Expected time to ruin

Figure 5.3: Ruin probability and expected time to ruin for different initial states.

We can see that the numerical results for the ruin probability look like exponentially decreasing functions for both the good and the bad initial state. We will later in this section plot the logarithm of the ruin probability to check whether that is a straight line, which would imply that the ruin probability is indeed an exponentially decreasing function. As we would expect, the ruin probability for the good initial state is lower than the ruin probability for the bad initial state. The results for the time to ruin again look like straight lines for both the good and the bad initial state. We can clearly see that the time to ruin for the good initial state is generally slightly higher than for the bad initial state. As the capital grows relatively quickly in the good state, the probability that ruin will happen in the beginning is very low. If ruin were to happen, it will most-likely not happen until a transition to the bad state has been made. When the Markov process starts in a bad state on the other hand, the probability that ruin will occur quickly is much higher. Therefore, the time to ruin when starting in a good initial state is generally higher than when starting in a bad initial state.

To see whether the ruin probability is indeed given by an exponentially decreasing function and, moreover, to check whether the exponent of the ruin probability when beginning in the good state, R_G is the same as the exponent of the ruin probability when beginning in the bad state, R_B, i.e. $\psi_G(x) = C_G e^{-Rx}$ and $\psi_B = C_B\, e^{-Rx}$, we will plot the logarithm of the ruin probabilities. We know that $log(\psi_G(x)) = log(C_G\, e^{-Rx}) = log(C_G) - R \cdot x$ and also $log(\psi_B(x)) = log(C_B) - R \cdot x$. So if the ruin probability is indeed given by an exponentially decreasing function, the logarithm of the ruin probability should be a linearly decreasing function. Furthermore, if the exponents are the same for the good and the bad states, the slopes of these linear function should be the same. The logarithm of the ruin probability for the same parameters can be seen in Figure 5.4.
It appears that these lines are indeed linear, implying the ruin probability is given by an exponentially decreasing function. However, even though in the book Ruin Probabilities [5] it is shown that the exponent of the ruin probability in a Markovian environment is the same for each initial state, these lines do not seem to be parallel, which would mean that the exponents of the ruin probabilities with different initial states are, in fact, not the same. This could be a consequence of the fact that we are considering a model with stochastic premium, whereas the computations in the book Ruin Probabilities are performed on the Cramér-Lundberg model. When we only consider the results after $x = 15$, the points do appear to be parallel, so it could also be a result of a numerical error, for example, caused by the way of simulating the state-switch. Further research is necessary to draw any conclusions on this matter.

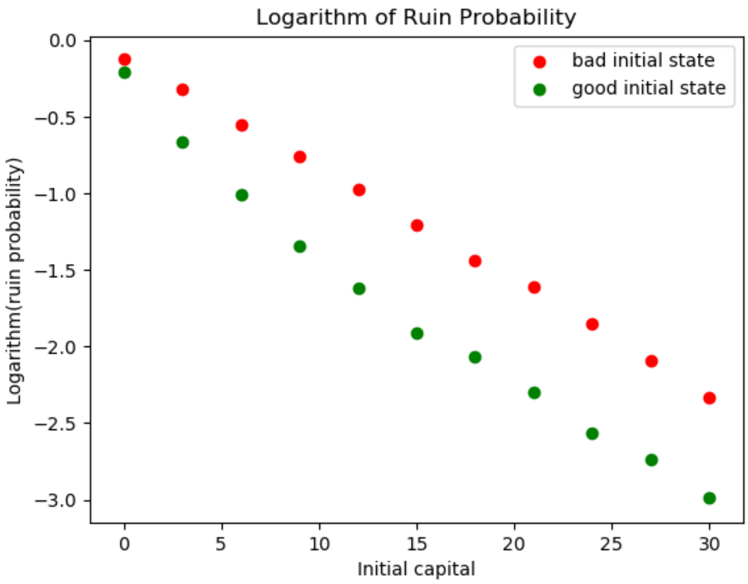


Figure 5.4: Logarithm of the ruin probability for both a good and bad initial state.

Now that we have taken a look at the ruin probability let us also take the time to ruin into consideration, as that remains the main objective of this report. The first thing we will look at is how the holding time affects the difference in the time to ruin between starting in a good state or in a bad state. Intuitively, this difference would become smaller when the holding time gets smaller and for sufficiently small H_i with $i \in \{G, B\}$, the difference will converge to zero. When visualising this, we use the same parameters for the premiums that have been used in the previous figures of this section. For the claims, let us take the following parameters: $\lambda_B^- = 1.5$ and $\mathbb{E}[Z_j]_B = 1.3$ for the bad state and $\lambda_G^- = 1$ and $\mathbb{E}[Z_j]_G = 1$ for the good state. We will first take $\gamma = \nu = \frac{1}{30}$ and then increase these parameters to $\gamma = \nu = \frac{1}{5}$, leaving all the other parameters unchanged.

Figure 5.5: The effect of the holding time on the difference in the time to ruin between starting in a good state or in a bad state.

In this figure, it can be seen that the difference in the time to ruin is relatively big when the holding times are chosen large, as expected. When the holding time decreases, so does the gap in the time to ruin for different initial states. However, for a smaller holding time, the numerical results of the time to ruin with both the good and the bad initial state are very close to the results with a large holding time and a bad initial state. In other words, it appears as if the holding time in both states does barely affect the time to ruin when the model starts in a bad state, but does affect the time to ruin when the initial state is the good state. This is remarkable, as we might expect that the results with small holding times would lie in between the results with larger holding times, i.e. the results for both initial states are equally affected by a different choice for the holding time. This might be a consequence of the possibly false assumption that not changing the scheduled claims has no effect on the outcome of the simulation, especially now the holding time is taken relatively small. To exclude this possibility, we will adjust the simulation so that scheduled claims are being changed and observe what the effect is on the results by only considering the case where the initial state is the good state and the holding time is still equal to $H_i = 5, \forall_{i \in \{B,G\}}$.

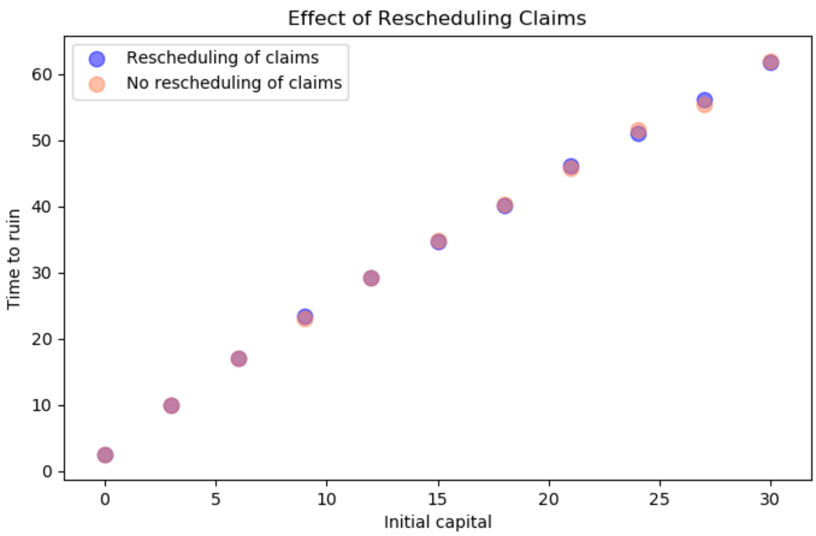


Figure 5.6: The effect of rescheduling the claims on the expected time to ruin.

As can be seen in this Figure 5.6, rescheduling the claims after a transition occurs right before the claim has no significant effect on the results for the time to ruin, confirming the assumption we made is fair. Hence, we can conclude that no numerical error has occurred and that, at least for this particular set of parameters, the holding time only significantly affects the results for the time to ruin when starting in the good state.

In the previous chapters, we have seen how the time to ruin is influenced when the type of premium and its parameters are being adjusted. We will now focus on what happens with the time to ruin when we change the parameters for the claims. Also, we will randomize the initial state according to the parameters of the Markov process. Hence, the probability we start in the bad state is equal to $\frac{\gamma}{\gamma+\nu}$ and the probability we start in the good state is $\frac{\nu}{\gamma+\nu}$. We are interested in what happens when the safety loading condition is kept the same but the difference between the good and bad state is being increased. We know that the safety loading condition looks as follows: $\frac{\nu}{\gamma+\nu}\left(\lambda_G^- \,\mathbb{E}[Y_i]_G\right) + \frac{\gamma}{\gamma+\nu}\left(\lambda_B^- \,\mathbb{E}[Y_i]_B\right) < \lambda^+ \,\mathbb{E}[Z_j]$. Hence, the safety loading factor is given by $\theta = \frac{\lambda^+ \,\mathbb{E}[Z_j]}{\frac{\nu}{\gamma+\nu}\left(\lambda_G^- \,\mathbb{E}[Y_i]_G\right) + \frac{\gamma}{\gamma+\nu}\left(\lambda_B^- \,\mathbb{E}[Y_i]_B\right)} - 1$. We will keep the rate at which the claims arrive constant and only consider changes in the size of the claims in both states and the holding time of each state. We get the following plot:

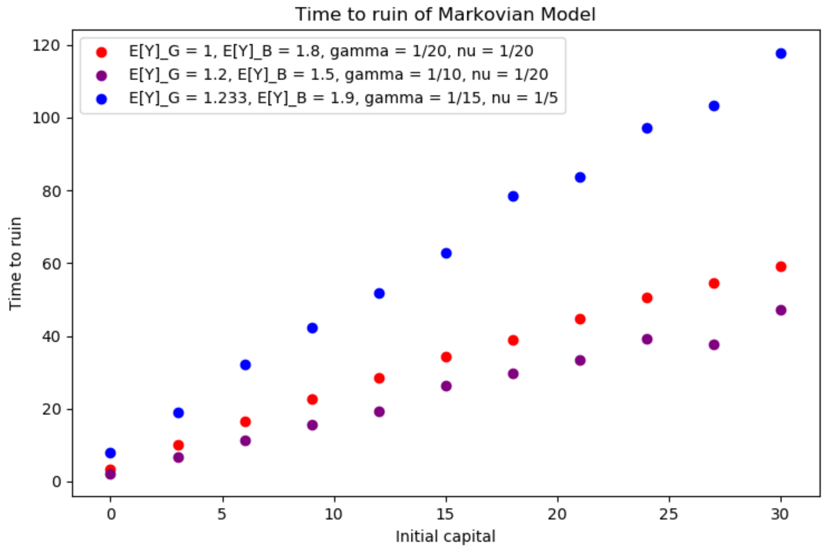


Figure 5.7: Time to ruin of the Markovian model for several claim sizes and holding times.

The red dots represent the results for the same parameters as in Figure 5.3, where the holding time of both states is the same and the difference between the expected claim sizes in both states is equal to $\mathbb{E}[Y_i]_G - \mathbb{E}[Y_i]_B = 0.8$. The purple dots represent the results for a set of parameters where the expected holding time of the bad state, $\mathbb{E}[H_B]$, is twice as large as the expected holding time of the good state, $\mathbb{E}[H_G]$, and the difference between the expected claim sizes in both states is only 0.3. These results for the expected time to ruin are quite similar to the results of the parameter set given by the red dots, only slightly lower and less accurate, implying the insurer's capital grows faster for this parameter set. The blue dots represent the results where the expected holding time of the good state is three times as large as the bad state and the difference between the expected claim sizes is approximately 0.667. It is remarkable that, even though the Markov process is in the good state most of the time, the expected time to ruin is significantly higher for this set of parameters than it is for the other two. This implies that the insurer's capital grows far less quickly when at some moments in time the safety loading factor is very low. Therefore, we can conclude that an insurance company should try to keep its safety loading condition large at all times, as this can contribute to its capital sooner reaching a point where ruin will not occur anymore.

6 Conclusion and discussion

In this study we have looked at four insurance models, all modelling the claims or the premiums in a different way. The first model, treated in Chapter 2, is known as the Cramér-Lundberg model or the classical Poisson risk model and has already been investigated extensively. An analytic expression for the expected time to ruin was already found in the book "Insurance Risk and Ruin" [2] and has been explicated step-by-step in this report. For the second model, the model with both stochastic premium and stochastic claims, we used the expression for the ruin probability which has been found in the bachelor end project of Sabine Geurts and computed a second order nonhomogeneous differential equation to find an expression for $\psi_1(x)$. With these two expressions we were able to find an expression for the expected time to ruin analytically. We also have shown that this model converges to the Cramér-Lundberg model when parameter limits are taken such that the stochastic premium converges to a linear premium. When doing the computations for the third model, treated in Chapter 4, we first computed the expected time to ruin for an auxiliary model with stochastic premium only and hyper-exponentially distributed premium sizes. When the expected time to ruin was found for this auxiliary model, parameter limits were taken such that the expression for the expected time to ruin of this model converged to an expression for the original model with both stochastic and linear premium. At the end of each of these three chapters, a stochastic simulation of the corresponding model has been performed. These simulations served as a way of validating the expressions that have been found, but also as a way of observing the effect of certain parameters on the time to ruin. In the last chapter, a model with stochastic premium and stochastic claims with an underlying Markov process has been investigated. Even though no expression for the time to ruin has been found analytically, we have seen how the model behaves for several sets of parameters by using stochastic simulation. The main conclusion we can draw from this simulation is that the safety loading factor in each state has a significant impact on the time to ruin.

This brings us to the first point of discussion of this project, namely the lack of an analytic expression for the expected time to ruin of the model with Markovian states. We now know that even though the ruin probability behaves like an exponentially decreasing function, it is not of the form $C \cdot e^{-Rx}$. Also, we have only looked into the case where the state space of the Markov process consists of two states. Further research could look at this model more extensively by using a different approach to finding the ruin probability and consider what happens when the state space is extended to n states. A second point of discussion is that in this report, we have only looked at exponentially distributed claim and premium sizes. Different distributions might result into different expressions for the time to ruin, hence future studies could look into expressions for the time to ruin independent of the claim or premium size distribution, i.e. in terms of the distribution functions. Furthermore, we saw that the numerical results for the time to ruin become less accurate when the ruin probability decreases. By simulating this differently, for example running the model until a certain number of ruins has been reached instead of running the model for a fixed number of runs, more accurate results for the time to ruin can be found. However, generating the results the way it has been done in this report allowed us to use the accuracy of the numerical results as an indication of the ruin probability.

Lastly, one might wonder what the social importance of this project might be. The expected time to ruin is in general a useful quantity for insurance companies as in a way it represents the insurer's capital growth. When for a given set of parameters the insurer's capital grows relatively slow, this will result into a higher expected time to ruin, as the probability of ruin occurring at a later point in time is relatively high. Furthermore it is useful for an insurance company to have insight in the expected time to ruin, given that ruin will occur at some point, so they know when to anticipate, for example, by collecting more premiums.

Bibliography

[1] A.V. Bolkov. "The Cramér-Lundberg Model with Stochastic Premium Process". In: *Theory of Probability and Its Applications* 47.3 (2006), pp. 489–493.

[2] David C.M. Dickson. *Insurance Risk and Ruin*. Cambridge, United Kingdom: Cambridge University Press, 2017.

[3] Sabine Geurts. "The ruin probability of insurance models with linear and stochastic premium". Bachelor's Thesis. Eindhoven University of Technology, 2019.

[4] Mark Richard Greene. *Insurance*. 2019. URL: https://www.britannica.com/topic/insurance.

[5] Asmussen Soren and Albrecher Hansjorg. *Ruin Probabilities (2nd Edition)*. Vol. 2nd ed. Advanced Series on Statistical Science Applied Probability v. 14. World Scientific, 2010. ISBN: 9789814282529.

7 Appendix

```
#Main simulation class
class Main:

    #Initialize the main simulation class
    def __init__(self, claimSizeDist, claimTimeDist, startCap):
        self.claimSizeDist = claimSizeDist
        self.claimTimeDist = claimTimeDist
        self.Premium = Premium
        self.startCap = startCap

    #Code for the Cramer−Lundberg model
    def CLsimulation(self, T, Premium):
        #Initializing parameters
        fes = FES()
        res = SimulateResults()
        t = 0
        Capital = self.startCap
        res.StoreCoordinates((t, Capital))
        #Schedule first claim
        FirstClaimSize = self.claimSizeDist.rvs()
        FirstClaim = Event(Event.CLAIM, t+self.claimTimeDist.rvs(), FirstClaimSize)
        fes.add(FirstClaim)
        while(Capital >=0 and t<T):
            #Jump to next event and update premium
            e = fes.next()
            Capital += (e.time − t)∗Premium
            t = e.time
            res.StoreCoordinates((t, Capital))
            if e.type == Event.CLAIM:
                #Substract claim amount from capital
                Capital −= e.amount
                res.StoreCoordinates((t, Capital))
                #Schedule new claim
                NewClaim = Event(Event.CLAIM, t+self.claimTimeDist.rvs(), self.claimSizeDist.rvs())
                fes.add(NewClaim)
        return t


    def Model2simulation(self, T, premiumSizeDist, premiumTimeDist, lamplus):
        fes = FES()
        res = SimulateResults()
        t = 0
        C = 0
        P = 0
        Capital = self.startCap
        ClaimSizes = self.claimSizeDist.rvs(T)
        #Generating all random variables at once to increase efficiency
        A = T∗lamplus
        PremiumSizes = premiumSizeDist.rvs(A)
        ClaimArrivalTimes = self.claimTimeDist.rvs(A)
        PremiumArrivalTimes = premiumTimeDist.rvs(A)
        FirstClaim = Event(Event.CLAIM, ClaimArrivalTimes[C], ClaimSizes[C])
        FirstPremium = Event(Event.PREMIUM, PremiumArrivalTimes[P], PremiumSizes[P])
        fes.add(FirstClaim)
        fes.add(FirstPremium)
        while(Capital >=0 and P<T−1 and C<T−1):
            e = fes.next()
            t = e.time
```

```python
            #Check whether next event is a claim or a premium
            if e.type == Event.CLAIM:
                Capital -= e.amount
                C+=1
                NewClaim = Event(Event.CLAIM, t+ ClaimArrivalTimes[C], ClaimSizes[C])
                fes.add(NewClaim)
            elif e.type == Event.PREMIUM:
                Capital += e.amount
                P+=1
                NewPremium = Event(Event.PREMIUM, t+ PremiumArrivalTimes[P], PremiumSizes[P])
                fes.add(NewPremium)
        return Capital , t



    def Model3simulation(self , T, premiumSizeDist , premiumTimeDist , lamplus , Premium):
        fes = FES()
        res = SimulateResults()
        t = 0
        C = 0
        P = 0
        Capital = self.startCap
        res.StoreCoordinates((t , Capital))
        ClaimSizes = self.claimSizeDist.rvs(T)
        A = T*lamplus
        PremiumSizes = premiumSizeDist.rvs(A)
        ClaimArrivalTimes = self.claimTimeDist.rvs(A)
        PremiumArrivalTimes = premiumTimeDist.rvs(A)
        FirstClaim = Event(Event.CLAIM, ClaimArrivalTimes[C], ClaimSizes[C])
        FirstPremium = Event(Event.PREMIUM, PremiumArrivalTimes[P], PremiumSizes[P])
        fes.add(FirstClaim)
        fes.add(FirstPremium)
        while(Capital>=0 and P<T-1 and C<T-1):
            e = fes.next()
            Capital += (e.time - t)*Premium
            t = e.time
            res.StoreCoordinates((t , Capital))
            if e.type == Event.CLAIM:
                Capital -= e.amount
                res.StoreCoordinates((t , Capital))
                C+=1
                NewClaim = Event(Event.CLAIM, t+ ClaimArrivalTimes[C], ClaimSizes[C])
                fes.add(NewClaim)
            elif e.type == Event.PREMIUM:
                Capital += e.amount
                res.StoreCoordinates((t , Capital))
                P+=1
                NewPremium = Event(Event.PREMIUM, t+ PremiumArrivalTimes[P], PremiumSizes[P])
                fes.add(NewPremium)
        return Capital , t



    def MarkovModel(self , T, gamma, nu, claimSizeDistbad , claimTimeDistbad , premiumSizeDist , premiumTimeDist , StatechangedistGoodtoBad , StatechangedistBadtoGood):
        fes = FES()
```

```
            res = SimulateResults()
            t = 0
            Capital = self.startCap
            res.StoreCoordinates((t,Capital))
            beginstate = Distribution(stats.uniform()).rvs()
            #Determine begin state: 1 is good, 0 is bad
            if beginstate <= nu/(gamma+nu):
                state = 1
                FirstStateChange = Event(Event.STATECHANGE, t+StatechangedistGoodtoBad.rvs())
                fes.add(FirstStateChange)
                FirstClaim = Event(Event.CLAIM, t+claimTimeDist.rvs(), claimSizeDist.rvs())
                fes.add(FirstClaim)
            else:
                state = 0
                #Schedule state change
                FirstStateChange = Event(Event.STATECHANGE, t+StatechangedistBadtoGood.rvs())
                fes.add(FirstStateChange)
                FirstClaim = Event(Event.CLAIM, t+claimTimeDistbad.rvs(), claimSizeDistbad.rvs())
                fes.add(FirstClaim)
            FirstPremium = Event(Event.PREMIUM, t+premiumTimeDist.rvs(), premiumSizeDist.rvs())
            fes.add(FirstPremium)
            while(Capital>=0 and t<T):
                e = fes.next()
                t = e.time
                res.StoreCoordinates((t,Capital))
                #Handle state change and schedule new one
                if e.type == Event.STATECHANGE:
                    state = 1 - state
                    if state ==1:
                        NewStateChange = Event(Event.STATECHANGE, t+StatechangedistGoodtoBad.rvs())
                        fes.add(NewStateChange)
                    elif state == 0:
                        NewStateChange = Event(Event.STATECHANGE, t+StatechangedistBadtoGood.rvs())
                        fes.add(NewStateChange)
                elif e.type == Event.CLAIM:
                    Capital -= e.amount
                    res.StoreCoordinates((t,Capital))
                    if state == 0:
                        NewClaim = Event(Event.CLAIM, t+self.claimTimeDist.rvs(), self.claimSizeDist.rvs())
                    elif state == 1:
                        NewClaim = Event(Event.CLAIM, t+claimTimeDistbad.rvs(), claimSizeDistbad.rvs())
                    fes.add(NewClaim)
                elif e.type == Event.PREMIUM:
                    Capital += e.amount
                    res.StoreCoordinates((t,Capital))
                    NewPremium = Event(Event.PREMIUM, t+premiumTimeDist.rvs(), premiumSizeDist.rvs())
                    fes.add(NewPremium)
            return t
#Set all variables and distributions
T = 1000
Premium = 1.8
startCap = 3
#Variables for good state
```

```python
lamMin = 1
ExpClaim = 37/30
claimTimeDist = Distribution(stats.expon(scale=1/lamMin))
claimSizeDist = Distribution(stats.expon(scale=ExpClaim))
#Variables for bad state
lamMinbad = 1
ExpClaimbad = 1.9
claimTimeDistbad = Distribution(stats.expon(scale=1/lamMinbad))
claimSizeDistbad = Distribution(stats.expon(scale=ExpClaimbad))
#Variables for premium
lamPlus = 1
ExpPrem = 2
premiumTimeDist = Distribution(stats.expon(scale=1/lamPlus))
premiumSizeDist = Distribution(stats.expon(scale=ExpPrem))
#Variables for State change
gamma = 1/15
nu = 0.2
stateChangeGoodtoBad = Distribution(stats.expon(scale = 1/gamma))
stateChangeBadtoGood = Distribution(stats.expon(scale = 1/nu))

nrRuns = 10000


#Generate results for larger number of runs and for different initial capitals
RuinTimes = []
RuinProbs = []

for j in np.arange(0,31,3):
    Results = []
    simu = Main(claimSizeDist, claimTimeDist, j)
    Results = []
    for i in range(nrRuns):
        tim = simu.MarkovModel(T, gamma, nu, claimSizeDistbad, claimTimeDistbad, premiumSizeDist, premiumTimeDist, stateChangeGoodtoBad, stateChangeBadtoGood)
        if tim < T:
            Results.append(tim)
    RuinProbs.append(len(Results)/nrRuns)
    RuinTimes.append(mean(Results))

print(RuinProbs)
print(RuinTimes)



#Event Class
class Event:

    CLAIM = 0
    PREMIUM = 1
    STATECHANGE = 2


    def __init__(self, typ, time, amount=0):  # type is a reserved word
        self.type = typ
        self.time = time
        self.amount = amount


    def Change_Time(self, newtime):
        self.time = newtime

    def __str__(self):
        s = ('Arrival', 'Departure')
```

```
        return s[self.type] + " of Car " + str(self.customer)+ ' at t = ' + str(
    self.time)

    def __lt__(self, other):
        return self.time < other.time

#Future Event Set class
class FES :

    def __init__(self):
        self.events = []

    def add(self, event):
        heapq.heappush(self.events, event)

    def next(self):
        return heapq.heappop(self.events)

    def isEmpty(self):
        return len(self.events) == 0
#Class used for plotting a single run
class SimulateResults:

    def __init__(self):
        self.coordinates = deque()


    def StoreCoordinates(self, coordinates):
        self.coordinates.append(coordinates)

    def getCoordinates(self):
        return self.coordinates


    def pltSurplus(self):
        x,y = zip(*self.coordinates)
        plt.figure()
        plt.plot(x,y)
        plt.title('Single Simulation Run of Model with Markov-modulated claims')
        plt.ylabel('Insurers Capital')
        plt.xlabel('Time')
        plt.ylim(0, max(y)+1)
```